Interactive learning in bioinformatics

Harald S. Fianbakken <haraldf@ifi.uio.no>

May 16, 2008

Abstract

Studies in natural science and mathematics are loosing popularity among young people all across the world. In Norway this is no different. This is a concerning matter for the future and ways of motivating students for these studies are appreciated. This thesis tries to find suitable topics from bioinformatics and develop an e-learning environment in order to motivate high school students for such studies. A user test is then performed in order to evaluate the e-learning environment developed.

Contents

1 Introduction **7**

1.1 Motivation 7

1.2 Purpose 8

1.3 Structure of this thesis 9

2 E-learning **11**

2.1 What is e-Learning? 11

2.2 Creating an e-learning environment 12

2.3 Selecting a pedagogical approach 15

2.3.1 Common pedagogical approaches 15

2.4 Guidelines for the development of an efficient e-learning environment 17

2.5 HCI principles 19

3 Bioinformatics and microarrays **22**

3.1 Bioinformatics 22

3.2 Microarray technology 24

3.2.1 Gene expression 25

3.2.2 Microarrays 27

3.2.3 What is a microarray? 27

3.2.4 How does microarrays work? 28

3.3 Microarrays in bioinformatics 30

3.3.1 Clustering algorithms 31
3.3.2 Clustering example 41

4 Survival analysis **45**
4.1 Parameters 45
4.2 Point estimators 46
4.3 Survival function and survival analysis 47
4.4 Kaplan-Meier estimator 47
4.4.1 Details of the Kaplan-Meier estimator 48

5 Study cases **53**
5.1 Terminology 53
5.2 Possible study cases 54
5.2.1 Gene expression and its relation to survival 55
5.2.2 The importance of looking at DNA / genes 57

6 A scenario for setting the objectives **59**
6.1 Purpose of a scenario 59
6.2 General design and implementation issues 60
6.3 Introducing the scenario 61
6.4 Things to consider 62
6.4.1 Identifying a disease 63
6.4.2 Selecting a disease to represent 65
6.5 The interaction - Step by step 66
6.6 Summary 68

7 Implementation **70**
7.1 Criteria 70
7.2 Tools and frameworks 71
7.3 Choosing a programming environment 72
7.4 Gui-components 78

7.4.1 Choosing a gui-toolkit . . . 78
7.4.2 JFreeChart . . . 81
7.5 Choosing an IDE . . . 82
7.6 Summary . . . 83

8 Application design **84**
8.1 General design . . . 84
8.2 Framework . . . 87
8.2.1 Model . . . 87
8.2.2 View . . . 89
8.2.3 Control . . . 90
8.3 The scenario . . . 90

9 User interface design **94**
9.1 General . . . 94
9.2 Constraints on user interface . . . 95
9.3 Content . . . 95
9.4 Navigation . . . 95
9.5 Content frames . . . 97
9.6 Consistency . . . 99
9.7 Application in use . . . 99
9.8 Interactive elements . . . 101
9.8.1 Microarrays . . . 102
9.8.2 Interactive dogma . . . 104
9.8.3 Creating plots . . . 104
9.8.4 Survival curves . . . 105
9.8.5 Clustering techniques . . . 105
9.8.6 Quizzes . . . 107

10 Reusability **108**
10.1 Application framework . . . 108

10.1.1 InteractiveStudyCase - A GUI controller interface . . 108
10.1.2 Controllers 109
10.2 Data containers - Models 109
10.2.1 SurvivalDataTable 109
10.2.2 Gene - An interface for representing genetic information 110
10.2.3 Microarray 110
10.3 Components - Views 110
10.3.1 MicroArrayView 110
10.3.2 InteractiveDogmaView 111
10.3.3 ButtonControlPanel 112

11 Evaluation of prototype **113**
11.1 Design of the experiment 113
11.2 Deployment 114
11.3 Feedback 114
11.4 Results of the experiment 115
11.5 Evaluating the results 116
11.6 Actions for improvements 117

12 A user test of the system **118**
12.1 Introduction 118
12.2 Design of the user test 118
12.3 Pre/-Post quiz 120
12.4 Deployment of the user test 120
12.5 Results 123
12.5.1 Pre-quiz 123
12.5.2 Post-quiz 125
12.6 Summary 127

13 Conclusion and future work **131**

13.1 Conclusion 131

13.2 Future work 133

13.2.1 Improvements 133

13.2.2 Ideas for new study cases 135

13.2.3 New interactive elements 136

Acknowledgments

I would like to thank my mentors Arne Maus and Ole Christian Lingjærde. Without their help, guidance and feedback I would never have completed this thesis.

Also, thanks to "Risør Videregående Skole" and "Oslo by Steinerskole" for helping me with the user test performed. A special thanks goes to the biology teachers for their positive attitude and for letting me use their classes for the user test.

I would like to thank fui (fagutvalget ved informatikk) for providing a coffee maker at the study room. The large amount of caffeine produced helped me through late nights. Finally I would like to thank all my friends for bearing with me through the process and listening to my complains when lacking motivation.

Chapter 1

Introduction

1.1 Motivation

Studies in natural science and mathematics are loosing popularity among young people all across the world. In Norway this is no different. The number of applicants for these studies in higher level education (universities/colleges) has been decreasing the past few years. According to the statistics from "Samordna opptak" in 2006, the numbers of applicants for such studies decreased by 9.7% from 2005 to 2006 [38]. In addition to this, only 20% of all the students at high school level chooses science courses as optional courses. As the report indicates these numbers cannot be explained only by a single factor. However it its a clear indication that science subjects are loosing popularity and this is a matter of concern. It is not only that recruiting new students to such subjects that is difficult, also a great deal of the students starting these studies drop out after the first two years, either because they lack an interest in the subjects or because they find them too difficult.

However, among these dark numbers there is still hope for recruiting students to science related subjects. It has been reported an increase in the number of students leaving secondary school to continue with high school

instead of the traditional work-related education. This can be promising as many of these students are likely to continue a higher education when they finish high school.

The latest statistics from "Samordna opptak" [39] shows a slight increase in the number of applicants to natural science subjects at the University of Oslo. When looking at a longer period of time the number of applicants has decreased significantly, so easing down because of this statistics would be wrong. The University of Oslo still wishes to recruit more students to science related subjects and ways to get the students attention towards this are appreciated.

The classical pattern where students go to lectures and watch the lecturer summarize a topic or they grab the book and read about a topic is not the best way to learn for everyone. Alternative ways of representing course material can be motivating and even aid some of the students in the learning process. Also, by motivating students and make them more interested in a topic, it can be used as a tool for recruiting students to further studies. This leads us to the purpose of this thesis.

1.2 Purpose

The purpose of this thesis is to come up with a selection of interactive study cases that can be used for triggering an interest for science among students. The study cases should also serve an educational purpose, not just as a motivating factor. They should give the students an insight in the topics they illustrate which hopefully will trigger their interests and making them want to learn more. Since this is a thesis in bioinformatics the study cases will primarily use topics that can be found within this field. E-learning (see chapter 2) is going to be the approach used for representing the study cases. One of the study cases I come up with will then be implemented,

creating an e-learning environment and a user test will be performed in order to evaluate the application developed.

As far as I know, there are no free, available frameworks for dealing with interactive case studies within biology/bioinformatics. Therefore, the idea would be to construct a framework for the study cases with a sense of flexibility so some of it can be re-used to illustrate other examples in bioinformatics. Hopefully, this will show how one can use computer interaction in a learning situation for handling biological information and how to represent fairly complex science topics in an understandable manner for students. Such a flexible tool might have the side effect of making it easy to apply in other fields of science.

The result of this study is not intended to replace any form of previous educational methods, but rather work as a supplement and as an aid and motivational factor for students at different levels. The target groups will aim for students at high school that are curious about science, and those students just started a higher level education in a science subjects. Hopefully this will draw some attention towards such studies and maybe help in recruiting students for future studies or help keeping already existing students motivated and "on track". Since people have different needs, this approach might not be ideal for everyone, but it is one of many possibilities.

1.3 Structure of this thesis

This section is intended to give a brief overview of the chapters in this thesis.

The next chapter (2) will introduce e-learning and how to implement an e-learning environment. The preceding two chapters (chapter 3 and chapter 4) will look at interesting topics that can be included in study cases

for the e-learning environment. Chapter 3 will introduce bioinformatics and present an interesting technology named microarrays. Chapter 4 will introduce survival analysis and some basic statistics surrounding this. The topics presented in these chapters will be used in order to come up with study cases for the e-learning environment.

Chapter 5 will come up with a couple of study cases that could be used for the e-learning environment and select one of them to develop. The following chapter 6 will concretize the study case further by coming up with an informal scenario and discuss the interaction involved. Chapter 7 will discuss the tools, frameworks and programming environment selected for implementing the e-learning environment (application).

The implementation of the application will be discussed in chapter 8 and in chapter 9. Chapter 10 will cover how some of the components and material developed can be reused in similar applications. An evaluation of the prototype developed will be performed in chapter 11 to catch potential bugs and design flaws. The final application are then tested and evaluated by performing a user test in chapter 12. The last chapter will summarize what has been accomplished, what could have been done differently and discuss future work.

Chapter 2

E-learning

As a result of the technological revolution we have had the past few decades, new learning methods have been developed. E-learning is a term that is frequently used and it is a direct result of the new technology that is offered to us. As indicated in chapter 1 an interactive application will be developed using the principles of e-learning. This chapter will give an introduction to what the term e-learning refers to and how to implement a user friendly and efficient e-learning environment so this can be used when developing an interactive application.

2.1 What is e-Learning?

There are many definitions of the term e-learning and not all of them are consistent with each other. To make it a bit more confusing there are many different terms used which is addressing e-learning. [1]

One thing that is common in most of the definitions on e-learning is that they all involve the use of technology for the aid in educational purposes. This captures the essential part of e-learning.

[1] Terms such as "Computer Aided instruction" (CAI) or "Computer aided learning" (CAL) are often used instead of e-learning, but they usually address the same domain

[43] stated that e-learning is: "The delivery of a learning, training or education program by electronic means. E-learning involves the use of a computer or electronic device (e.g. a mobile phone) in some way to provide training, educational or learning material".

The problem with definitions such as this is that it says little of what it is and how we can use it. In order to use e-learning for the purpose of this thesis, a more solid understanding of the term is needed.

2.2 Creating an e-learning environment

In order to create an e-learning environment one must:

- Establish a clear primary goal and objective for the environment.
- Decide what technologies to use
- Decide a pedagogical approach

In order to make an efficient e-learning environment, these decisions must be made early in the development process. Establishing the primary goal and the objectives early is essential for the remaining steps involved. The primary goal and objectives for the application as a whole were settled in chapter 1. As we shall see later there are sub-goals and objectives as a result of the study case included that also needs to be defined. These sub-goals and objectives are defined along the process in creating the study case and not in advance such as the primary goal established in the introduction.

Some technologies have proven to be quite efficient and are more frequently used than others in e-learning today. The following list describes some interesting technologies that can be used for an e-learning system [24] :

- **Dynamic web based applications**

  Dynamic web based applications is a fairly recent technology because

of some of the improvements in the web technology. Flash, Web 2.0 and Silverlight (Microsoft TM) are some of the technologies that open up for interactive web applications to be developed. They have the advantage of being able to run on most computers through a web browser without any hazel. However, there are still problems related to web applications and e-learning. One common problem is the difference in how web browsers act. The web application can be displayed (and act) differently depending on what browser the user is running. The development on web standards has helped in this situation, and most new web browsers behave similar today. It is worth noting that there is still a large amount of old web browsers being used and when developing an e-learning environment to be exposed to people outside the initial test environment, this must be taken in account. Also, web based applications have some limitations when it comes to advanced interaction. The interaction is usually a bit more difficult to develop than for a normal application. As the technology improves some of these disadvantages are likely to disappear.

- **Virtual classrooms**

  Virtual classrooms is a technology that allows for a tutor or mentor to guide a group of students and have discussions. This collaborative environment can be compared to a normal classroom where both the students and the tutor are active.

- **Virtual blackboard**

  A virtual blackboard is an interactive application where the content changes dynamically and the user can control (normally) what to see. This is the classic definition we think of when we hear interactive applications. However, virtual blackboards can be much more. A

fairly recent technology that falls under this term, is a technology that allows for a remote mentor (over distance, e.g. online) to draw or add content onto a virtual screen dynamically which is displayed to the users. This is in many cases very similar to a classic blackboard but can be much richer in content. The content can include multimedia such as movies, images and sound. Often virtual classrooms use such a virtual blackboard.

- **Games**

  Games can be both entertaining and educational. There are many different types of games; some aim for the collaborative environment and others focus more on the learning objectives. 3D games (Such as Second life [8]) are often used when referring to e-learning 2.0. Using this in e-learning has become more popular as the technology has improved. Games offer many interesting possibilities for learning. Integrated collaborative environments, sharing ideas and multimedia and being able to communicate live are some of the many features offered. Other, simpler games (2D or text-based games) can be used and can be useful. Simpler games can have the advantage over 3D games, in the ease of use, complexity and because of the fact that they are easier to develop and require less time. Games can in some cases introduce a competitive factor which for some are motivating.

The indented e-learning environment will be a combination of a virtual blackboard and a textual game. Further details are discussed in chapter 6 where I come up with a scenario for the application and roughly sketch how the application should work. Next we must consider what kind of electronic devices the e-learning system will run on. E.g.: Are we developing a web-application that is supposed to run on all mobile devices and computers or is it a standalone virtual "blackboard" that runs on a PC? The limitations of the technological platform must be accounted for.

The technological platform may limit our choices for interaction (and visa versa). The last thing to think of when developing an e-learning system is what pedagogical approach to use.

2.3 Selecting a pedagogical approach

The pedagogical approach taken when developing an e-learning environment is often limited by the goals and objectives set out, but there can be room for creativity when choosing between the different approaches. The next section will describe some of the most common pedagogical approaches that are used in order to see if any of them can apply for the e-learning environment being developed.

2.3.1 Common pedagogical approaches

- **Instructional design**

  This is perhaps one of the most common used approaches to e-learning up till now. [26] gives the following definition of instructional design: "Instructional design models typically specify a method, that if followed will facilitate the transfer of knowledge, skills and attitude to the recipient or acquirer of the instruction."

  With other words, this approach focuses on creating a method (set of instructions) for achieving goals. Normally, a 5 step model is used for creating instructional material. The model is called ADDIE (Acronym for the 5 phases) and contains the following steps:

  - Analyze - analyze learner characteristics, task to be learned, etc.
  - Design - develop learning objectives, choose an instructional approach
  - Develop - create instructional or training materials

- Implement - deliver or distribute the instructional materials
- Evaluate - make sure the materials achieved the desired goals

The ADDIE model is in many ways similar to the traditional waterfall method [42] used in software development.

- **Social constructivism**

  The major focus of social constructivism is to uncover the ways in which individuals and groups participate in the creation of their perceived understanding of a problem [40]. Collaborative environments are required for this approach. Social constructs are generally understood to be the product of human choices, rather than getting the understanding of a problem by reading some material. One of the problems with this approach is how to design the material to present to the group. It has to be concise enough for the group to be able to discuss it and come up with the right solution, yet vague enough to make the students have a discussion (or in a larger setting, delegate work and do some research to find a solution).

- **Contextual perspective**

  This is one of the approaches that are gaining the most popularity within e-learning environments today. Contextual perspective takes "the better" of the two previous approaches and merges them into one approach. [13] gives the following definition: "Contextual perspective focuses on the environmental and social aspects which can stimulate learning. Interaction with other people, collaborative discovery and the importance of peer support as well as pressure"

I will primarily be using an "instructional design" approach, because of the fact that it is easier to design an applicable scenario and the boundaries that I have set for this thesis are better suited for this approach. However,

using a contextual perspective would be really interesting and should be considered if developed within a larger timeframe. When describing and developing the e-learning system, I should be able to cover the phases of ADDIE.

2.4 Guidelines for the development of an efficient e-learning environment

Consider the problem of how to implement an efficient e-learning environment. To deal with this problem, I will look at some guidelines for developing an e-learning environment.

Bohannon et al. give in their article on "Design Principles for Online Instruction" [16] a list on how to implement an efficient e-learning environment. For simplicity I have included only those of particular relevance to this thesis:

- Develop the course around clearly defined learning objectives and goals, and clearly communicate these to the learners. The targeted groups and primary goal for the thesis has already been mentioned and will not be discussed further. However, additional learning objectives and sub-goals will be discussed later in chapter 5 where different study cases that could be used are presented.

- Special attention must be given to how online courses are displayed. Artistry is not the goal. Instead, focus on organization to allow ease of navigation and learning enhancement. Graphics should present information to support learning. Attention must be given to student skill levels and equipment limitations when embedding audio, video, and web links. The KISS (Keep it simple, stupid) principle is important here. Don't overload the user with a fancy user

interface that is just this. Focus on developing screens that aim for supporting the learning material presented.

- Create a collaborative community spirit by requiring sharing activities between students and teachers, ensuring constructive criticism, maintaining motivation, and providing assessment tools with timely feedback.

- Keep the learning environment flexible. Individual needs, interests, and objectives must be considered, but should not become the end in itself. Knowledge must be built on in real-time and customized to meet educational goals.

Most of these principles will apply for the application with some rephrasing and interpretation of the statements.

Creating a collaborative community spirit will be difficult in this case. This does not apply for this e-learning environment, but should be considered in similar environments because it allows the students to communicate and give feedback.

Item 2 and 4 in Bohannons list is also very relevant. Since the application is interactive, it is important to think through how to organize the information being displayed and to make an effort in creating the UI (user interface) as intuitive and flexible as possible for the students to use. It is always a difficult question to answer what a good user interface is.

The overall guidelines on how to develop an e-learning environment has been covered and it is time to look at design principles concerning the user interface and interaction details. More specific I will look at the HCI (Human computer interaction) principles that can help in developing a better and more user friendly application and that describes the best practices in developing user interfaces.

2.5 HCI principles

There are numerous guidelines that describe good HCI. Shneiderman [41] has captured the most important principles which I will keep in mind and follow when designing the application. Only principles of particular relevance to this application have been included.

Shneiderman's Principles of Human-Computer Interface Design

- **Strive for consistency**
  - Consistent sequences of actions should be required in similar situations
  - Identical terminology should be used in prompts, menus, and help screens
  - Consistent color, layout, capitalization, fonts, and so on should be employed throughout.
- **Offer informative feedback**
  for every user action, the system should respond in some way (for example, a button will make a clicking sound or change color when clicked to show the user something has happened). Especially when the application needs time to finish a request, we need to show the user that something is happening. Normally, displaying a loading screen of some kind would solve this. This is a critical point for heavy operations, because the users loose interest in the application if they don't get any feedback at all while waiting for a request.
- **Design dialogs to yield closure**
  - Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions shows the user their activity has

completed successfully. Showing the user that this is a multistep procedure motivates them to continue. And keeping it simple by not trying to include all the actions on one screen makes it less scary for the user.

- **Offer error prevention and simple error handling**
  - Design the form so that users cannot make a serious error; for example, prefer menu selection to form fill-in and do not allow alphabetic characters in numeric entry fields
  - If users make an error, instructions should be written to detect the error and offer simple, constructive, and specific instructions for recovery
  - Segment long forms and send sections separately so that the user is not penalized by having to fill the form in again - but make sure you inform the user that multiple sections are coming up
- **Permit easy reversal of actions**
  Provide undo buttons where it is applicable. In situations that allows for mistakes to be made, this is crucial for the user experience. In situations where special considerations are done to avoid mistakes, this is less important (e.g. the users only have one choice or the user is notified that the current action cannot be undone).
- **Reduce short-term memory load**
  - A famous study from 1956 suggests that the amount of information a person can remember from one exposure is between 5 and 9 elements [32] . Most guidelines suggest that a maximum of 7 elements should be used (Millers magic number). The essential part is to reduce short term memory load by considering the interface design. This can be done by screens where options

are clearly visible, or using pull-down menus, icons and other visual aids.

The application being developed should aim to follow most of these principles.

Chapter 3

Bioinformatics and microarrays

3.1 Bioinformatics

To see what possibilities for interactive study cases that lays within the field of bioinformatics, we have to know what bioinformatics is all about. This section give an introduction to bioinformatics, domains of usage and why this is such an important field of science.

I will discuss some of the definitions others have made for the term "bioinformatics" and try to get an understanding of bioinformatics that can be used throughout the thesis.

Both in biology and in computer science the development has been exceptional the past few decades. The discovery of the DNA, the HUGO project mapping the entire human genome and the availability of computational power (Now, even Quad-core CPUs are available for desktop computers) to mention some of it, opens for ways of research one could hardly have imagined some years ago. Along with the new technology came large amounts of biological data that needed (and still needs) to be stored, analyzed and viewed. Using the computational power we have today we can develop quite sophisticated methods solve this. The

field that arose from combining biology and computer science is what is we usually think of when we hear the term bioinformatics.

NCBI (National Center for Biotechnological Information) tries to describe bioinformatics by using the following definition: "Bioinformatics is the field of science in which biology, computer science, and information technology merge into a single discipline".

This is clearly a very wide definition which tries to fetch as much information in one sentence as possible and it is easy to find examples that could fit into this definition yet not be related to what bioinformatics is intended to capture. E.g. a botanist that collects data for the plants and uses a computer to store this would fit into this definition but is normally not what we think of when we hear bioinformatics. The definition needs to be narrowed down and be more specific in order to understand what bioinformatics focuses at. Also, NCBI's definition does not say anything on what bioinformatics aim to solve, what methods to use nor if other science disciplines are involved. We could ask ourselves: "What are we analyzing and what techniques do we use to extract the information?". NCBI's definition on bioinformatics lacks the basic disciplines of the other sciences it applies for processing the data, such as mathematics and statistics. Without these disciplines, bioinformatics would be simply just to store and display biological data.

The website Bioinformatics.org has a different definition and approach when describing bioinformatics that answers some of these questions: "Bioinformatics use mathematical, statistical and computational methods for solving complex problems such as analyzing and sequencing biological information" [12]. This definition is getting closer to something we can relate to. Sequencing and analyzing biological information can be misinterpreted, but normally it relates to the molecular part of biological information (genetic sequences, DNA, gene-expressions). Yet,

this definition is overwhelming so dividing the problems into distinct parts is needed. Luckily, NCBI has done this neatly and they have divided bioinformatics into three domains that will be used in describing bioinformatics [35]:

- The development of new algorithms and statistics with which to assess relationships among members of large data sets.
- The analysis and interpretation of various types of data including nucleotide and amino acid sequences, protein domains, and protein structures; and
- The development and implementation of tools that enable efficient access and management of different types of information.

This is not a research project in the manner of developing new algorithms; I will aim at using already known algorithms and statistical methods. One could argue that some of the visualization needed for the interactive application could be new algorithms, but they are more correctly viewed as new ways of representing known algorithms. Now that we have a brief understanding of what bioinformatics is about, we will look at an example of a technology used in biology and how bioinformatics work with the data this technology provide.

3.2 Microarray technology

An important field within bioinformatics is microarray analysis. The microarray technology generates large amounts of data that needs to be stored, analyzed and compared. The microarray technology will be explained in order to understand what the data represents and to see how it is used. As for the informatics part I will go through some common algorithms used when analyzing the data produced. The

algorithms, techniques and problems surrounding microarray and analysis are complex topics and could be a thesis on its own. Therefore, a selection of suitable algorithms will be described for the intention of re-using some of the material in a study case that suited for the targeted group of this e-learning environment.

The discussion on how microarray data can be stored and organized has been left out from this thesis. Such an example could be suitable in a different context with a different target group or motivation (e.g. targeting informatics students).

3.2.1 Gene expression

In order to understand how microarrays work and are used, we need to know a little bit about how genes work. The cells in our body contain identical genes, but the different cell types differ in many other ways and suit different purposes (e.g. skin cells, nerve cells). How can they do that if they all share the same genes?

The expression (level of activity) of the genes varies between cell types (and even within a cell type) and over time. We may think of a gene as being active ("on") or inactive ("off") at a particular time in a particular cell. In order to understand how a gene can be turned “on” and “off” we need to understand the central dogma of biology (see figure 3.1). DNA is transcribed into mRNA, which again is translated into proteins. It is often written DNA->RNA->Protein. As we should remember from elementary biology proteins perform many critical functions in our cells and they are partly responsible for the way our cells work. I will not go into further details on how proteins specifically work, but for those interested [37] provides more details about proteins and their functions.

There are some exceptions to the central dogma; e.g. we can manipulate the process in a laboratory and retro viruses break the central dogma by

injecting pieces of RNA directly into DNA using an enzyme called reverse-transcriptase. This can have critical consequences for an organism since the DNA is altered and new (different) proteins will be transcribed instead of the original ones. [37].

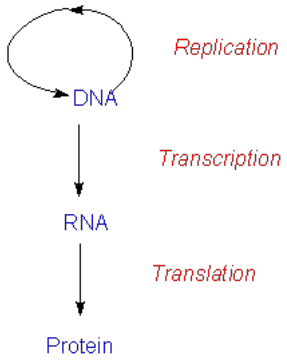


Figure 3.1: The central dogma of biology. DNA is replicated, transcribed to RNA and translated into protein.

Gene expression is a very complex and regulated process. A cell can respond to both environmental changes or to its own needs. It does this by regulating the amount of proteins being produced in the cell. When a cell is producing small amounts (below a given threshold) of a certain protein the gene may be said to be turned "off"; otherwise it is turned "on" (Expressed). So when talking about gene expression we refer to the amount of proteins produced. There are different levels of expression depending on the amount of mRNA produced (coding for a certain protein). The reason for this is, is because mRNA is translated into a protein (central dogma). Therefore, the cell regulates the amount of protein produced by regulating the transcription process in the cell from DNA to mRNA. This process is called volume control [19]. Note that the process in regulating the gene expression is not quite understood and lot of work remain in understanding gene regulation. There can be other processes regulating

gene expression that we might not be aware of yet.

The next section will describe the technology called microarrays and we shall see how we can use this for designing experiments to measure gene expression.

3.2.2 Microarrays

Microarrays are important for today's biological research. This technology has opened for a range of new experiments we had been unable to do earlier. [18] shows an example on how microarrays are used in current research. Here they show how different types of DLBCL (Distinct large B-cell lymphoma) was identified by using microarrays. This is one of many examples on how microarrays are used in today's research. [1]

Shi illustrates the importance of DNA microarrays: "DNA Microarrays - A technology that is reshaping molecular biology" [27].

He is referring to the many possibilities the DNA microarrays opened for. Traditional methods in molecular biology used to work with a very small set of genes in one experiment. With the old methods getting the whole picture of a gene function (and co-operating genes) was hard, if not impossible to do. The next section will give a brief introduction to the microarray technology and describe how they can be used. In the end we see how this technology relates to bioinformatics.

3.2.3 What is a microarray?

A microarray is a small glass (or silicon chip) that can contain thousands of gene-sequences. A microarray can be used to measure gene expression within a single cell, e.g. see how the gene expression vary over time in the respective cell, or it can be used compare gene expressions in two different

[1] There are different types of microarrays and that they have different usage. DNA microarrays is one of the many types available

cell types (e.g. a healthy cell and a diseased). The domains of usage for microarrays are many. If we want to survey a large number of genes or when we have a small sample for study a microarray can be useful. This technology is a major advance because we are able to make gene expression profiles for thousands of genes in different cells and compare them. Microarrays as a technology is still considered to be in it's infancy and there are many possible studies that might not have been explored yet [19].

3.2.4 How does microarrays work?

The process of creating a microarray is a multistep procedure that will be described briefly. For those particularly interested, further details on each step involved can be found on [22]. The basic steps in the process of creating a microarray is shown in figure 3.2.

The steps in creating a microarray are as follows:

1 **Extract RNA, make cDNA:** RNA is extracted from both sample and reference cells (E.genes. a sick and a healthy cell). The enzyme reverse transcriptase is used to make a complementary DNA (cDNA). The cDNA is a single strand that contains the complementary nucleotides to the template RNA it was created from. Note that not all RNAs are transcribed with the same efficiency to cDNA. This is called a reverse transcriptase bias, and as a result, two genes might not give the same result when we are measuring gene expression [33]. This is a problem we must be aware of in our experiments with microarrays. We must have a reference cDNA and the sample cDNA in our experiment.

2 **Label sample with fluorescent dyes:** Florescence molecules are used to label the cDNA. The reference cDNA is often labeled with Cy3 (giving a green color), and the sample is labeled with Cy5 (giving

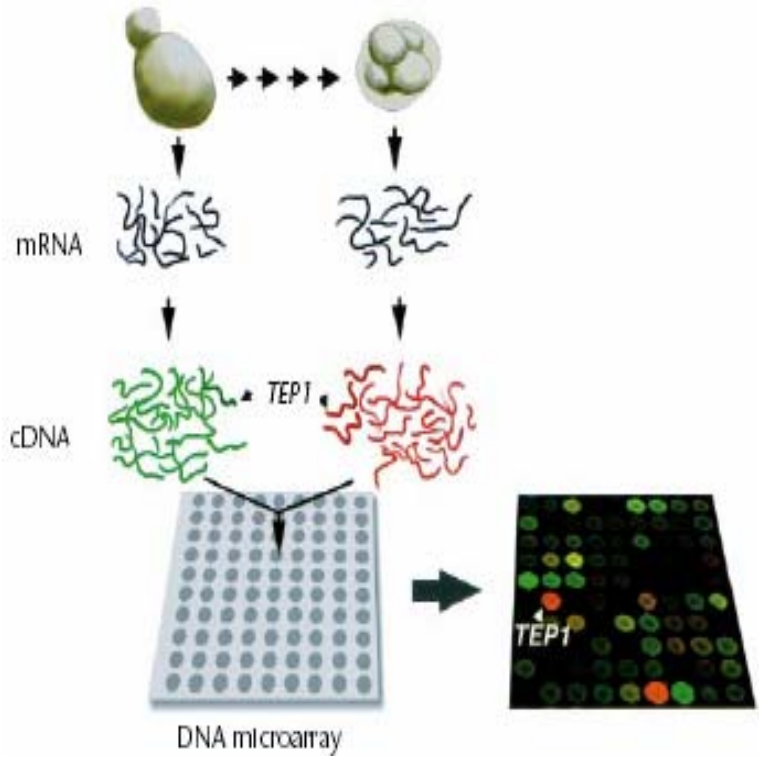


Figure 3.2: The steps in making a microarray [6]

a red color). This is important when they later hybridize to the array and we want to measure the amount hybridized.

3 **Hybridize to microarray:** The two labeled cDNA samples are mixed and flushed over the microarray. They will bind together if one of the samples contain cDNA which is complementary to the DNA sequence in a given position in the array (This is called hybridization). Every spot on the microarray should contain enough DNA for both samples to be able to hybridize without interfering with each other.

4 **Scan intensity and make image:** The microarray can now be used to extract data. In order to do so, we need to create an image of

the microarray and analyze the spots. By running the microarray through a special laser such that the laster affects every spot on the array, we can read the Cy5 and Cy3 signals separately. Two images (one for each fluorescent) are produced. The images are then merged together, giving one image with the two intensities mapped (See figure 3.3). From this image we can extract numeric data for each spot and perform an analysis.

5 **Analyze data:** Analyzing the data is the difficult part. The data we are dealing with is complex and the structure of it is often not quite understood. In addition, there are many sources of error that can affect the result, which we need to take in account (Dust on the microarray, irregular spots, background noise etc.). The data extracted from a microarray experiment are intensities for each gene on the microarray. The data is often stored as a log ratio of the sample and reference in question (see figure 3.4).

A microarray experiment can be designed to look at the gene expression profile over different patients, different cells or to see how the gene expression varies with time. As mentioned earlier, the domains of usage are many. These experiments produce massive amounts of data that needs to be analyzed. The next section describe where bioinformatics fits into all of this and will show how clustering can be used for analyzing microarray data.

3.3 Microarrays in bioinformatics

Microarrays would be of less value without bioinformatics and algorithms that are capable of doing the analysis required to get useful information from the microarrays. The vast amount of data generated needs to be clustered, aligned and analyzed. Good algorithms for matching and

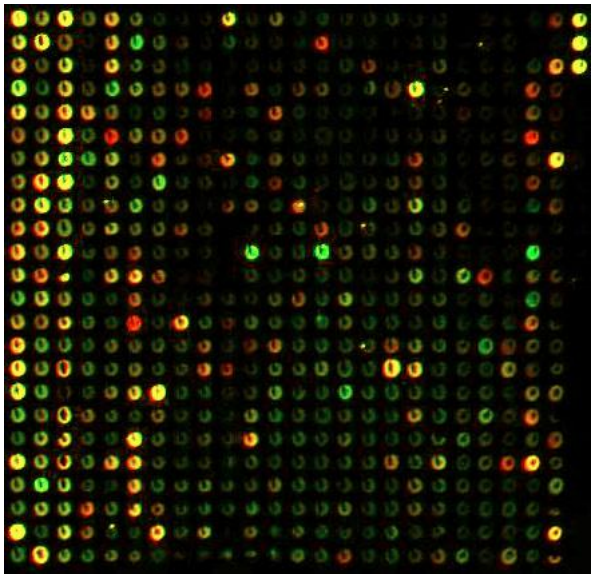
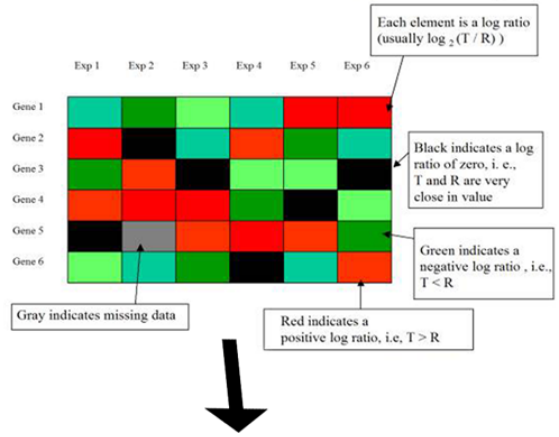


Figure 3.3: The intensity and color of each spot encode information on a specific gene from a sample [46]. A green spot indicates that the reference sample is the one being the most expressed. A yellow spot indicate that both samples are expressed just as much, whilst a red spot indicates that the sample has the strongest expression of the gene. If a spot is black, then no samples has that gene expressed

comparing profiles and special designed algorithms for experiments are needed. I will here go through some of the common techniques used when clustering and analyzing microarray data in order to find a an example suitable from informatics to include in the e-learning environment.

3.3.1 Clustering algorithms

Clustering is used to group similar entities (e.g. individuals, patients, genes) together and look at its relationship. Often, the result of a clustering algorithm is a visual image that is a very useful overview of the data involved (Such as figure 3.5). It is important to understand that these algorithms are used in many different settings and is not limited to



	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
Gene 1	-1.2	-2.1	-3	-1.5	1.8	2.9
Gene 2	2.7	0.2	-1.1	1.6	-2.2	-1.7
Gene 3	-2.5	1.5	-0.1	-1.1	-1	0.1
Gene 4	2.9	2.6	2.5	-2.3	-0.1	-2.3
Gene 5	0.1		2.6	2.2	2.7	-2.1
Gene 6	-2.9	-1.9	-2.4	-0.1	-1.9	2.9

Figure 3.4: Example on how a microarray image can be represented as numeric data. Each column represents all the gene expression levels from a single experiment, and each row represents the expression of a gene across all experiments. Each element is a log ratio. The log ratio is defined as $log_2(T/R)$, where T is the gene expression level in the testing sample, R is the gene expression level in the reference sample. [4]

microarray analysis that is the context used here.

Clustering of microarray data is a very essential part of microarray analysis. As stated above to cluster data is to partition the data into groups where they are similar in some manner. Similarity is a criterion we decide when choosing a clustering method. One important step in any clustering is to select a distance measure that determines how the similarity of two data elements is calculated. There are many different distance measures available. What algorithm and distance measure to choose is not always obvious and can vary from experiment to experiment. Data is often clustered multiple times, using different measures in order to see how the

result varies. I will here go through some of the most common distance measures. Later in this chapter an example will be constructed showing how a small data set can be clustered.

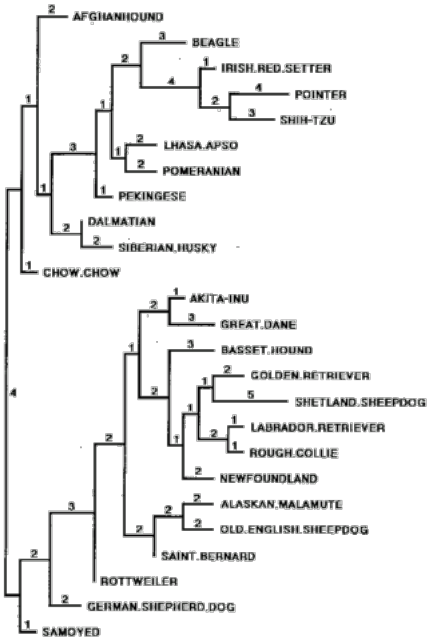


Figure 3.5: Dendrogram [1] - common way of representing the result of a cluster algorithm.

Distance measures

- **Euclidean distance**

  Euclidean distance is perhaps the distance measure that people are most familiar with. It is the normal distance one can measure when drawing a straight line between two points. The euclidean distance between two points $\underline{x} = (x_1, ..., x_n)$ and $\underline{y} = (y_1, ..., y_n)$ in an N-Euclidean space is defined as:

  $d(\underline{x}, \underline{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2}$

Consider the points $\underline{x}$ = (1, 2, 3) and $\underline{y}$ = (2, 4, 6). Using this formula the distance would be

$$d(\underline{x}, \underline{y}) = \sqrt{(1-2)^2 + (2-4)^2 + (3-6)^2} = \sqrt{1+4+9} = \sqrt{14}$$

- **Manhattan distance**

  Manhattan distance measures the distance between two points by using the absolute difference between the coordinates of the points. Manhattan distance is also known as taxicab geometry and city block distance. The manhattan distance between two points $\underline{x} = (x_1, ..., x_n)$ and $\underline{y} = (y_1, ..., y_n)$ is defined as:

  $$d(\underline{x}, \underline{y}) = |x_1 - y_1| + |x_2 - y_2| + ... + |x_n - y_n|$$

  Again consider the two points used in Euclidean distance $\underline{x}$ = (1, 2, 3) and $\underline{y}$ = (2, 4, 6). By using Manhattan distance we get the distance between the points:

  $$d(\underline{x}, \underline{y}) = |1-2| + |2-4| + |3-6| = 1 + 2 + 3 = 6$$

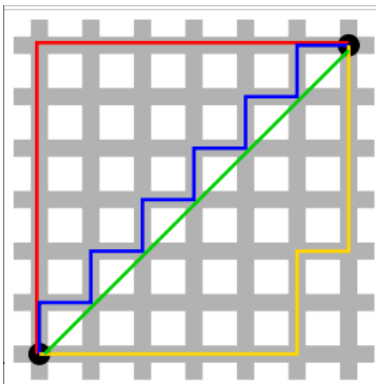


Figure 3.6: The difference between Euclidean and Manhattan distance [50]. The red, blue and yellow line here are equal in distance and is defined as Manhattan distance (or taxicab) while the green line is shorter and is the Euclidean distance between these two points. See how the two distances differ.

- **Pearson's correlation coefficient**

  Eucledian and Manhattan distance is widely used, but for gene expression profiles where variation between genes are very small, the profiles can be very difficult to differentiate [51].

  Correlation indicates the strength and direction of a relationship between two variables. In general statistical usage, correlation refers to the departure of two variables from independence. A number of different coefficients are used for different situations. One of the most common coefficients is the Pearson product-moment correlation coefficient which is obtained by dividing the covariance of the two variables by the product of their standard deviations [**?**]. Pearsons correlation coefficient is also frequently used in microarray analysis and clustering. Pearsons correlation can be defined as:

  c($\underline{x}$, $\underline{y}$) = $\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}(x_i-\bar{x})^2\sum_{i=1}(y_i-\bar{y})^2}}$, where

  $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

  Using the correlation, the distance is calculated using:

  d($\underline{x}$, $\underline{y}$) = $1 - c(\underline{x}, \underline{y})$

  Again consider the two points $\underline{x}$ = (1,2,3) and $\underline{y}$ = (2, 4, 6) from the two previous examples. Using the formula for Pearsons correlation for this example we get:

  c($\underline{x}$, $\underline{y}$) = $\frac{(x_1-\bar{x})(y_1-\bar{y})+(x_2-\bar{x})(y_2-\bar{y})(x_3-\bar{x})(y_3-\bar{y})}{\sqrt{(x_1+\bar{x})^2+(x_2+\bar{x})^2+(x_3+\bar{x})^2}\sqrt{(y_1-\bar{y})^2+(y_2-\bar{y})^2+(y_3-\bar{y})^2}}$.

  We calculate the average values: $\bar{x} = \frac{1+2+3}{3} = 2$ and $\bar{y} = \frac{2+4+6}{3} = 4$

  Inserting this in the formula gives us the correlation:

  c($\underline{x}$, $\underline{y}$) = $\frac{(1-2)(2-4)+(2-2)(4-4)+(3-2)(6-4)}{\sqrt{(1-2)^2+(2-2)^2+(3-2)^2}\sqrt{(2-4)^2+(4-4)^2+(6-4)^2}} = 1$

  From this, we can calculate the distance by subtracting the correlation

from 1:

$d(\underline{x}, \underline{y}) = 1 - c(\underline{x}, \underline{y}) = 0$

The correlation is always a number ranging from -1 to +1 (hence the distance ranges from 0 to +2). When the genes to be compared have the exact same expression pattern, the correlation coefficient between the profiles will be +1 and the resulting distance would be 0. When the genes have different expression, e.g. one gene is up-regulated and the other down regulated, the correlation will be -1 [51].

The choice of the distance measure can sometimes make a big difference in the final result.

Clustering algorithms

Apart from selecting a distance measure, different approaches are used to cluster data. Clustering algorithms are roughly separated into hierarchical and partitional algorithms. These two categories will be described here showing the difference between them and some of their properties.

Hierarchical clustering Hierarchical algorithms find successive clusters using previously established clusters. This is an easy approach and is quite intuitive to understand. The result of such an algorithm is usually a dendrogram that describes the relationship and distances between clusters (as shown in figure 3.5). There are two main types of hierarchical clustering; a top-down approach and a bottom-up approach:

– **Agglomerative clustering** :

  This technique is a top-down approach. All nodes start a separate clusters. It then merges the most similar nodes

forming a new node and traverses from top to bottom (usually generating some form of tree). The pseudo code describing an agglomerative method would be:

  * Initialize all data elements to be a separate cluster
  * Create a distance matrix between the clusters using some distance method.
  * Select the minimum distance found in the distance matrix between two clusters
  * Merge these clusters and calculate the new distance matrix.
  * If there are more than 1 cluster remaining, go to step 3.

- **Divisive clustering** :

  Divisive clustering on the other hand is a bottom-up approach. It starts with all data elements in one node. It then splits the cluster successively into smaller clusters, step-by-step until all clusters consist of a single data element. The approach is quite similar to the agglomerative clustering, except that it starts with the root node and successively splits the cluster using some minimizing criteria. Divisive clustering algorithms are seldom used in microarray analysis so I will not discuss it any further.

After clusters have been split or merged, hierarchical techniques need a way to calculate the distance between the new clusters and create the new distance matrix. There are 3 frequently used distance methods when calculating the new distance in a hierarchical clustering method [29]:

- **Single linkage** is defined as the minimum distance found between data elements in the two clusters (see figure 3.7). This is referred to as closest neighbors. Using this method often causes the chaining phenomenon [23], which is a direct consequence

of the single linkage method tending to force clusters together due to single entities being close to each other regardless of the positions of other entities in that cluster .

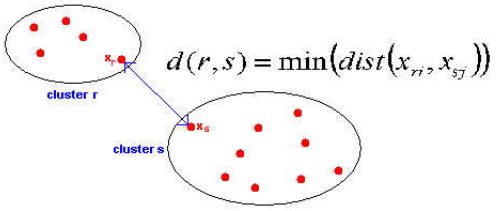


Figure 3.7: Single linkage is defined as the minimum distance between any two points in two clusters [5]

– **Complete linkage** is the opposite of single linkage. The distance is defined as the maximum distance that can be found between the data elements in the two clusters (see figure 3.8). This method should not be used if there is a lot of noise present in the data set. It tends to produce compact clusters. This method is useful if one is expecting entities of the same cluster to be far apart in multi-dimensional space (provided there is no noise). In other words, outliers are given more weight in the cluster decision.

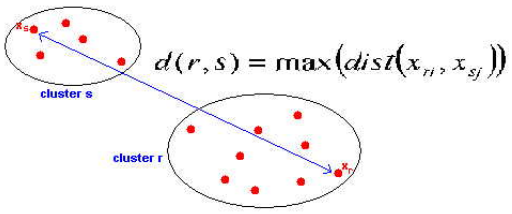
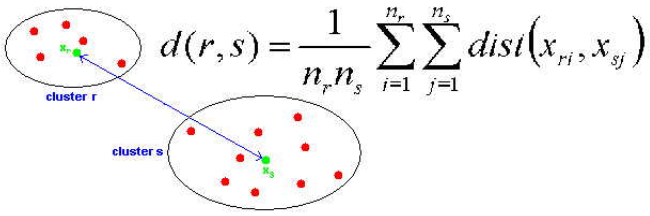


Figure 3.8: Complete linkage is defined as the maximum distance between any two points in the two clusters [3]

– **Average linkage** uses the mean distance between all possible

pairs of entities of the two clusters in question as the distance (see figure 3.9). It is therefore more computationally expensive than the two previously mentioned methods. The chaining problem is not observed for this method and outliers are not given any special favor in the cluster decision, which makes this method popular.



$$d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist\left(x_{ri}, x_{sj}\right)$$

Figure 3.9: Average linkage is defined as the average distance between all points in the two clusters [2]

Partitional (Non hierarchal) clustering While Hierarchal clustering methods successively splits or join clusters stepwise, partitional clustering methods tries to decompose the data set into disjoint sets directly. It does so by partitioning the data (thereby the name partitional clustering). These methods increase the strength of a cluster as more elements are assigned to be a member of a certain cluster. Partitional (Non hierarchal) clustering refers to all other clustering algorithms that are not hierarchal. In microarray analysis, K-means clustering is a common partitional clustering algorithm used.

The key points in partitional clustering are [11]:

- Each cluster will have a seed point and all objects within an initial distance are included in that cluster.

- Each data elements are assigned to the closest seeding point at a given iteration.

Partitional clustering algorithms are implemented very differently. To illustrate this, I have included a list showing how the iterations in partitional clustering may differ [11]:

- **1)** the sequential threshold (based on one cluster seed at a time and membership in that cluster fulfilled before another seed is selected, i.e., looping through all points before updating the seeds. K-MEANS clustering uses this approach)
- **2)** parallel threshold (based on simultaneous cluster seed selection and membership threshold distance adjusted to include more or fewer objects in the clusters, i.e., updating the seeds as you go along. ISODATA clustering method applies this approach)
- **3)** optimizing (same as the others except it allows for reassignment of objects to another cluster based on some optimizing criteria).

Also, the selection of seed points differ (even within the same algorithm). See the following list on how K-means clustering have different approaches in selecting seed points [34]:

- Let k denote the number of clusters to be formed.
- Fix k. (Later you can try k-1, k+1, etc.). We can choose k "seed" points to get started.
- The result depend upon the initial seed points, so often clustering is done several times, starting with different seed points. The k initial seeds can be :

* the first k cases
* a randomly chosen k cases
* k specified cases
* chosen from a k-cluster hierarchical solution.

In order to illustrate how to use clustering and mapping it to bioinformatics, a small example has been constructed. The example will be using hierarchical clustering, single linkage and manhattan distance.

The constructed example will manually cluster a data set step-by-step and show an agglomerative approach. The hierarchical clustering method shown here is intuitive and could be integrated in the e-learning environment.

3.3.2 Clustering example

Assume that we have measured the gene expression for four patients using a microarray and that we have the initial data shown in table 3.1 [2]. We think that the 4 genes shown in this example are related to types of lymph cancer. Patient 4 has an already known subtype, and we would like to see if any of the other patients could have a profile closely related. This example shows one way we can use cluster analysis and microarray data to answer such a question. This example will cluster the patients using a hierarchical agglomerative method using manhattan distance and single linkage.

The first step in the cluster algorithm is to calculate the initial distance matrix. Using the microarray data in table 3.1 a distance matrix can be calculated. To illustrate how this is done, the distance between patient 1 and patient 2 is calculated (using manhattan distance) here:

[2] An actual experiment is likely to contain more genes than this example

Recall the definition of manhattan distance:

d($\underline{x}$, $\underline{y}$) = $|x_1 - y_1| + |x_2 - y_2| + ... + |x_n - y_n|$

$\underline{x}$ = Patient 1 and $\underline{y}$ = Patient 2.

We read the expression for each gene from table 3.1 and compare the two patients:

d(Patient 1, Patient 2) = $|60 - 20| + |71 - 81| + |84 - 65| + |91 - 10| = 152$

The same method has been applied for all the patients and the resulting distance matrix is shown in 3.2.

The next step in the algorithm is now to select the minimum (shortest) distance. From the initial distance matrix (figure 3.2) we select this distance; in this case this is between patient 1 and patient 4 (distance

Table 3.1: The microarray data for the clustering example. The values here represent the gene expression for the different patients involved. For simplicity a scale ranging from 0 (not expressed) to 100 (highest expression) are used

	Patient #1	Patient #2	Patient #3	Patient #4
Gene 1	60	20	49	54
Gene 2	71	81	90	30
Gene 3	84	65	41	70
Gene 4	93	10	30	75

Table 3.2: Step 1. Initial distance matrix calculated using Manhattan distance

	Patient 1	Patient 2	Patient 3	Patient 4
Patient 1	-	152	166	79
Patient 2	-	-	82	155
Patient 3	-	-	-	139
Patient 4	-	-	-	-

of 79). Proceeding with the next step in the cluster algorithm these patients will merge and form a new cluster (Patient1 & Patient4).

A new distance matrix must be calculated. Since the new cluster contains more than one patient and we are using single linkage to calculate the new distance matrix we get: d("Patient 1& Patient 4", Patient 3) = min(d(Patient 1, Patient 3), d(Patient 4, Patient 3)). The new distance matrix with the merged clusters is shown in 3.3.

Table 3.3: Distance matrix. New distances calculated after merging patient 1 and patient 4 using single linkage, manhattan distance

	Patient 1&Patient 4	Patient 2	Patient 3
Patient 1&Patient 4	-	152	139
Patient 2	-	-	82
Patient 3	-	-	-

We repeat the previous step. Using the distance matrix 3.3 we select the next minimum distance, between Patient 2 and patient 3 (Distance of 82). Patient 2 and patient 3 will now merge in order to form a new cluster (Patient 2& Patient 3).

The last step in the cluster algorithm should be obvious since there are only two available clusters. The two clusters will merge and form the root-node (last cluster, containing all the patients). For simplicity the calculation of the last step has been left out but are easily obtained by using the same steps as before. The result of this clustering is shown as a dendrogram in figure 3.10.

This example has shown how clustering of microarray data can be used to see the relationship of gene expression profiles between patients. Imagine that we have 100 patients with gene expression values at our disposal and that they all have an unidentified subtype

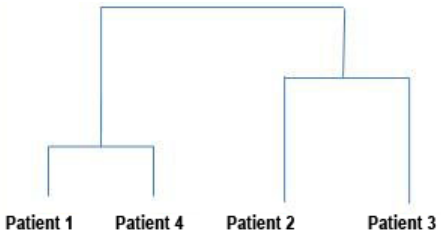


Figure 3.10: Dendrogram of the result from clustering example 3.3.2. The cluster was created using a hierarchical, agglomerative cluster algorithm

of cancer. By using a cluster algorithm as shown we could see if any of the patients have gene expression profiles that are related or if we have a profile for a specific type of cancer we can test the relationship for the other patients to compare it and see if they relate (this could mean that the patient most likely would have this type).

In other experiments it could be more interesting to see how gene expression vary over time and the clustering could be performed on time intervals instead of patients. Including a topic about microarrays in a study case would not be too difficult because of the many types of experiments we can design.

Chapter 4

Survival analysis

The chapter will give a brief introduction to the basic statistics of survival analysis. Most of the topics here are quite complex in nature and could be a thesis on their own. It would be out of scope to describe them in detail and including hard mathematical proofs. Therefore, this chapter is only intended to roughly cover the basics that are needed in order to understand survival analysis and see how we can use this in a study case later. Because of the importance of the Kaplan-Meier estimator and because it is suited for use in such a study case it will be covered in some more detail.

4.1 Parameters

A parameter is a function of the distribution of X, taking values in a parameter space A. Parameters are quantities that define certain characteristics of systems or functions [47]. When evaluating the function over a domain or determining the response of the system over a period of time, the independent variables are varied, while the parameters are held constant. The function or system may then

be reevaluated or reprocessed with different parameters, to give a function or system with different behavior. I will not go into any further details on parameters here because it can be quite complex and we will not be needing any further understanding of parameters later, but they are essential and worth mentioning briefly.

4.2 Point estimators

An estimate is a result of using a function (estimator) on some observable sample data. Hence, the estimator is used to estimate an unknown population parameter. There are many different estimators possible for any given parameter. Some criterion is used to choose between the estimators, although it is often the case that a criterion cannot be used to clearly pick one estimator over another. To estimate a parameter of interest (e.g., a population mean, a binomial proportion, a difference between two population means, or a ratio of two population standard deviation), the usual procedure is as follows [45]:

- Select a random sample from the population of interest.
- Calculate the point estimate of the parameter.
- Calculate a measure of its variability, often a confidence interval.
- Associate with this estimate a measure of variability.

More formally explained: point estimation involves the use of sample data to calculate a single value (known as a statistic), which is to serve as a "best guess" for an unknown population parameter.

4.3 Survival function and survival analysis

A survival function is a function denoting the probability of an event (death or failure) occurring later than some time t. S(0)=1 under normal circumstances unless there are instant deaths occurring in the data. A survival function can never increase with the time t. E.g. the following property will always be true for a survival function: S(u) <= S(t), where u > t. These functions are used in the branch of statistics that we call survival analysis or reliability theory (In engineering often named reliability analysis). Survival analysis attempts to answer questions such as [48]:

- what is the fraction of a population which will survive past a certain time?
- Of those that survive, at what rate will they die or fail?
- Can multiple causes of death or failure be taken into account?
- How do particular circumstances or characteristics increase or decrease the odds of survival?

There are many approaches to how to answer these questions. [20] shows how regression modeling of censored data can be used in survival analysis. I will not go into this method for survival analysis but instead look at an easier way of estimating survival, using the Kaplan-Meier estimator. This method is not as complex in nature and might be more suited to use in the e-learning environment here.

4.4 Kaplan-Meier estimator

Kaplan and Meier's paper (1958) is frequently referenced in many fields of science. Most disciplines of science use the Kaplan-Meier

estimator (KME) in some experiments today. The Kaplan-Meier estimator is also known as the "product limit estimator" (PLE) [30]. The intention with this section is to give an introduction to this estimator and show how to use it in experiments.

In experiments where we observe occurrences of events (that is in our interest; e.g. death, failure, etc.) the data can often be incomplete. Imagine an experiment where we observed 100 patients that has been given a treatment and the event of interest here is death. Some of these patients dropped out from the study (censored) for various reasons [1]. We would like to estimate the proportion whose lifetime would exceed a time (e.g. months, years) given the treatment. The censored data should not interfere with the estimate. This is the problem the Kaplan-Meier estimator (KME) was aims to solve [25].

Stated simpler, the KME is an estimator that can take censored and truncated data in account. Also, it is a useful estimator when the number of cases is small but representative and the exact time of an event is known. The next section will show the statistics behind the Kaplan-Meier estimator and illustrate how to use it by constructing two examples.

4.4.1 Details of the Kaplan-Meier estimator

The formal definition of the Kaplan-Meier estimator is as follows:

$S(t) = \mathbf{\Pi}_{\mathbf{t}_i <= t}(1 - \frac{\mathbf{d}_i}{\mathbf{R}_i})$

$S(t)$ the estimated survival function at time t

$\mathbf{\Pi}_i$ denotes the multiplication of the survival time of the event across all cases less than or equal to t

[1] Maybe the patient was relocated and unable to continue study or the patient died for a reason not of interest

$\mathbf{d}_i$ denotes the number of uncensored events at time t

$\mathbf{R}_i$ = number in case study - (number of events (censored+normal) prior to time t)

To show how this works and to show that it is possible to use it for something other than survival analysis a small example have been constructed. Assume that we wanted to create a Kaplan-Meier plot of the following experiment: The time after a group of shoplifters have been released from jail until they commit shoplifting again. Assume we have the following data set in this experiment: X (23, 21, 25, 20+, 13+, 17, 12, 10, 5) (+ indicate that the data was censored). Also, the size of the test group is set to 50. We order the data in an ascending order of time, and put up the cumulative function S(t) in order to create a KME plot. This is shown in table 4.4.1. This data is then used to create the Kaplan-Meier plot as shown in figure 4.1.

Data set 1

Events (time)	# $\mathbf{r}_1$	# c	# $\mathbf{r}_2$	# d	$\mathbf{s}_1$	S(t)
0	50	0	50	0	$\frac{50}{50}$	1
5	50	0	49	1	$\frac{49}{50}$	$1 * \frac{49}{50} = 0.98$
10	49	0	48	1	$\frac{48}{49}$	$0.98 * \frac{48}{49} = 0.96$
12	48	0	47	1	$\frac{47}{48}$	$0.96 * \frac{47}{48} = 0.94$
13	47	1	46	0	$\frac{46}{46}$	$0.94 * \frac{46}{46} = 0.94$
17	46	0	45	1	$\frac{45}{46}$	$0.94 * \frac{45}{46} = 0.92$
20	45	1	44	0	$\frac{44}{44}$	$0.94 * \frac{44}{44} = 0.92$
21	44	0	43	1	$\frac{43}{44}$	$0.92 * \frac{43}{44} = 0.90*$
23	43	0	42	1	$\frac{42}{43}$	$0.90 * \frac{42}{43} = 0.88$
25	42	0	41	1	$\frac{41}{42}$	$0.88 * \frac{41}{42} = 0.86$

#$\mathbf{r}_1$ number at risk prior to time t

#c number censored in interval

#$\mathbf{r}_2$ number at risk at end of time t

#d number that died at time t

#$\mathbf{s}_1$ the surviving proportion time t

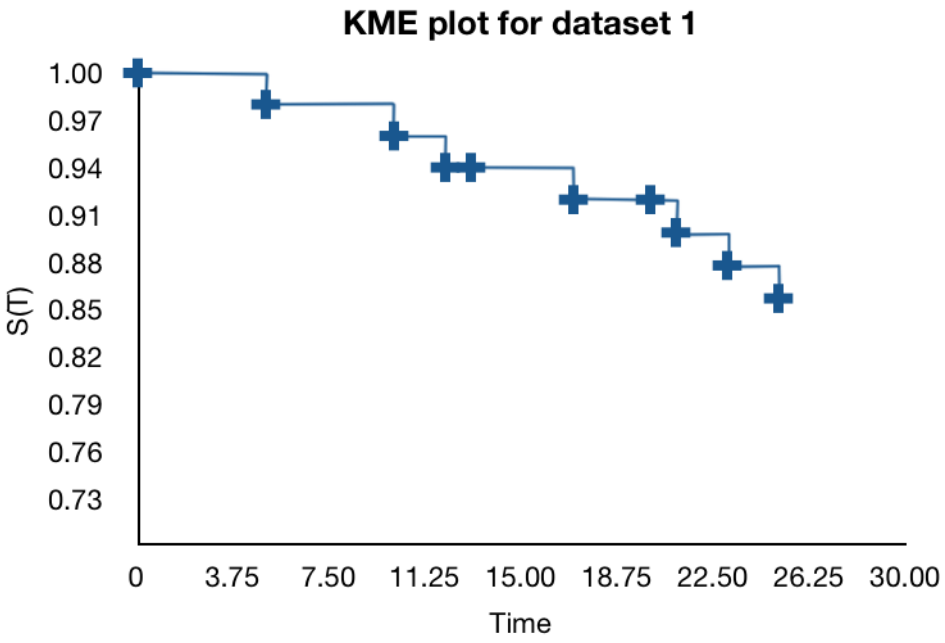


Figure 4.1: The plot for data set 1.

Assume there is another group of shoplifters and that they have been given hypnotherapy in order to prevent them from stealing after being released from jail. We are interested to see if the hypnotherapy had any affect on this group. For this we have the following data set X2 (22+, 22, 19, 15, 22, 15, 21, 16, 16, 14, 16, 18+, 12, 12+, 6, 11) (+ indicate that the data was censored). The same procedure as in the previous example is done (ordering the data and putting up a table with S(T)). This is done in table 4.4.1. From this data, we can create another Kaplan-Meier plot. This is easily combined with the first data plot in the same graph. The result of this is shown in figure 4.2.

Data set 2

Events (time)	# $\mathbf{r}_1$	# c	# $\mathbf{r}_2$	# d	$\mathbf{s}_1$	S(t)
0	70	0	70	0	$\frac{70}{70}$	1
6	70	0	69	1	$\frac{69}{70}$	$1 * \frac{69}{70} = 0.99$
11	69	0	68	1	$\frac{68}{69}$	$0.99 * \frac{68}{69} = 0.97$
12	69	1	67	1	$\frac{67}{69}$	$0.97 * \frac{67}{69} = 0.96$
14	67	0	66	1	$\frac{66}{67}$	$0.96 * \frac{66}{67} = 0.94$
15	66	0	64	2	$\frac{64}{66}$	$0.94 * \frac{64}{66} = 0.91$
16	64	0	62	2	$\frac{62}{64}$	$0.91 * \frac{62}{64} = 0.88$
18	62	1	61	0	$\frac{61}{61}$	$0.88 * \frac{61}{61} = 0.88$
19	61	0	60	1	$\frac{60}{61}$	$0.88 * \frac{60}{61} = 0.87$
21	60	0	59	1	$\frac{59}{60}$	$0.87 * \frac{59}{60} = 0.86$
22	59	1	57	1	$\frac{57}{58}$	$0.86 * \frac{57}{58} = 0.84$

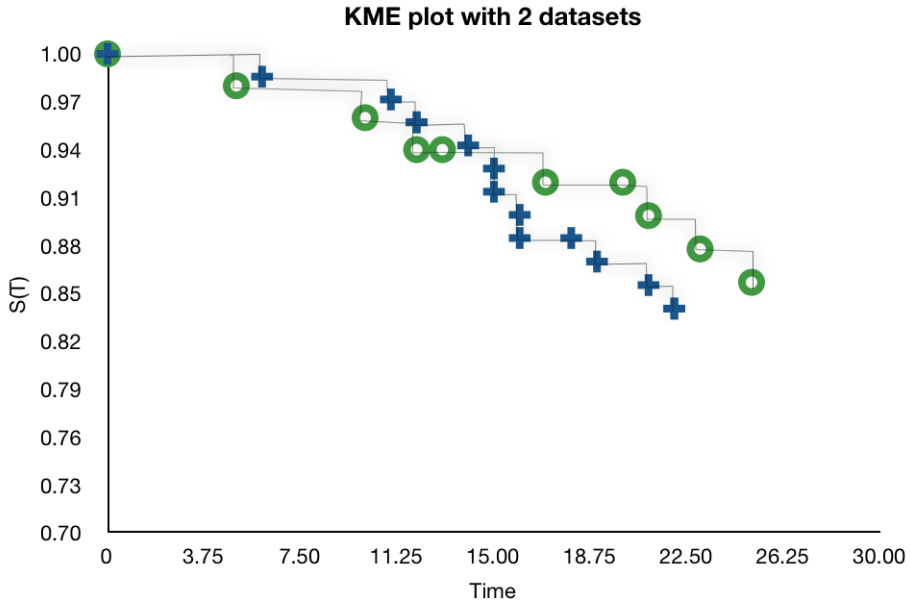


Figure 4.2: A Kaplan-Meier estimator plotted with two data sets.

A log-rank test, which is a hypothesis test used for comparing two survival distributions [14], would for this plot show that the difference in the two data sets is not significant. Making a conclusion based upon the results in this case would therefore be impossible, but the example should show how the Kaplan-Meier estimator can be used to test a hypothesis from data sets with missing (censored) data. The Kaplan-Meier estimator is fairly easy to understand without extensive knowledge of statistics, also the fact that it is used in many daily situations makes it an excellent topic to include in a study case.

Chapter 5

Study cases

5.1 Terminology

A study case is a collection of goals and objectives and a description of what we would like to achieve. The study case will be a guideline for the e-learning environment. It is a rough description of what we like to illustrate, possible interaction with the application and will establish the sub-goals and objectives within the e-learning environment. This definition however is quite overwhelming and would include too much for the implementation. Because of the limited time frame it will be narrowed down and focus on specific elements. To accomplish this we divide a study case into different scenarios with different approaches. The study case as a whole can still be used when developing a larger application with a larger time frame.

A scenario is an informal story (See chapter 6) which help in describing the detailed goals and learning objectives of the study cases we choose to include.

Because a scenario can be represented with different ways of

interaction and slightly different methods to achieve the goals we set, we divide a scenario into approaches.

It is not necessary for all approaches to cover all of the topics; but each approach should focus on at least one of the learning goals and topics that are set for the study case. By creating this terminology, it is possible to make specific implementations that have a narrow and specified learning focus. See figure 5.1 to see how the terms are connected together. The terms "Scenario" and "approach" are described more in detail in chapter 6 where they are used. This chapter will be focusing on study cases.

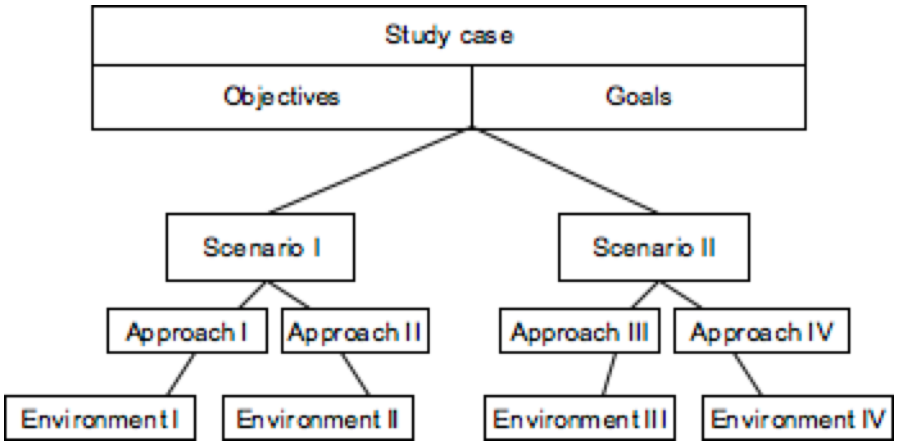


Figure 5.1: Terminology and how they are they are connected. Note: that in this application the focus is on developing one of the approaches for one scenario.

5.2 Possible study cases

At this point one should have an understanding of the terminology used and be familiar with bioinformatics and some of the topics that can be used in study cases. This leads us to the main purpose of this chapter which is to present some study cases that can be used as a

foundation for the development of the e-learning environment. Two examples of such study cases will be presented to illustrate that there are possibilities. The discussion of the study cases can be viewed as a part of the analyze and design phase (AD) of ADDIE described in chapter 2. One of the study cases will be selected and the preceding chapters will go through the process of developing that study case to a working example covering the other phases of ADDIE (develop, implement and evaluate). Further comments on other potential study cases are given in chapter 13.

5.2.1 Gene expression and its relation to survival

Description The main goal of this study case is to show that genes are expressed at different levels and that changes in the expression of genes may be associated with disease condition and reduced survival. The purpose is furthermore to show how microarrays can be visualized and changes in gene expression (of certain genes) can be directly mapped to a disease.

Interaction The users should see a simplified expression microarray and be able to adjust the expression by clicking a gene. The simplest form of interaction (E.g. beginner) can only switch the expression "on/off", while in a more advanced interaction one can choose between states. Once a gene expression has changed there should be a visual representation of the survival of an individual (or a group). This will be built upon a statistical model (Cox or Kaplan-Meier); and can be represented in various forms depending on the target group. E.g. graphs of different kinds or an easier form such as showing a population with one head per individual. Some other steps of interaction could involve:

- **Select disease**

  Select a disease to be represented by its genes in the microarray. The user clicks on a button and selects a disease from the internal database. The microarray is then redrawn with a representation of genes and data, where the students can adjust the expression of the genes. The microarray will be showing the expression under the conditions of the selected disease. We should make a note here that adjusting expression of a gene not related to the disease might also affect survival.

- **Create disease**

  This option could allow the student to create an own "disease" by selecting from a gene-pool, and creating an expression profile for a disease (e.g. by adjusting how much each gene should matter). The student can then try out his/her disease as an interactive case. This option is not meant to be realistic but rather a "fun" feature which would aim to give the students a better understanding on how genes and expression can be connected with survival.

- **Use real data**

  This option would use real data for estimation of parameters in the survival model. One problem with using real data is the representation of the microarray. If real data is to be used, it can be thousands of genes affecting the survival rate. Hence it would not be convenient to show all of these genes to the student using it. "Real data" will therefore be an intermediate with reading the data and calculating a hazard based on the "most important" genes, and displaying these genes in the microarray. A good statistical model calculating the hazard should be designed if this option is to be included in a study case.

Learning goals To summarize this study case, the learning goals here would be:

- Microarrays
- Gene expression
- Genes
- Survival analysis - and statistical models

5.2.2 The importance of looking at DNA / genes

Description This study case should show the users the importance of looking at genetic information when it comes to health related issues. The study case would be focusing on how important it is to look at examples such as gene-expression patterns, mutations or suppressed genes.

Interaction There are many approaches one could take that would aim for the main goal. This study case could include a lot of interactive elements to illustrate topics such as Kaplan-Meier survival curves, DNA and genes related to diseases and gene-expression. One idea is to see how microarrays can be used to match profiles for different diseases or to distinguish subtypes of a disease from each other. Other possible interaction would be to manipulate the DNA for specific genes to see how it would regulate gene-expression.

Learning goals Some of the learning goals are dependent on the scenario we set for the study case, but in most cases would include:

- Microarrays
- Gene expression

- Genes
- Survival curves - Kaplan-Meier

See that both these study cases share the same learning goals, but the intention and elements of interaction for the study cases differ. The latter study case has a lot of interesting elements and I will develop this further. The next chapter will be using this study case and take one approach in how to design this study case. This is done by coming up with a scenario that specifies the study case further and describes the interaction more in detail.

Chapter 6

A scenario for setting the objectives

6.1 Purpose of a scenario

To motivate the students into using the application and to make the interaction more interesting for the students, a scenario will be introduced. The scenario will work as a foundation for the selected study case. It is an informal story that will be the setting the students are introduced to, and it should give the students a purpose for using the application being developed.

The scenario will help setting the domain (establishing some guidelines). It will help describing what the students are supposed to learn and come up with a suggestion on how they will accomplish this. So the scenario suits two purposes; it aids the development and creativity process and the story itself will give the students motivation (a purpose of using the application different from "click, learn and read").

There are many good scenarios one could use. To focus on

one implementation and to make it as complete and thorough as possible, one scenario has been selected and details concerning that scenario are described. Before introducing the scenario, it is worth mentioning some important questions concerning the design and implementation.

6.2 General design and implementation issues

The application should provide an opportunity for the students to learn the topics introduced in the selected study case in details and let them learn this at their own initiative. The solution to this problem will be by introducing a topics guide in the application so that the students can choose themselves how detailed information they would like on the given topic, or they can simply choose to ignore it. People learn differently and we should at least introduce the two common general ways of learning; "read, learn, do, evaluate" and "do, read, learn, evaluate" (Theoretical approach vs. Practical approach - Some prefer reading to get the knowledge and some prefer getting the understanding by doing). In the implementation, the students are forced into clicking through the topics in a sequential manner, but they can choose to ignore the material and learn by doing instead.

Also in an approach where the students can read and learn before they interact, a pre-quiz can in many cases be a good idea. A pre-quiz could be used to give the students an idea of what they should learn, or it could be used to gather information for some experiment. A pre-quiz in this scenario does not seem suitable because the students can choose to ignore reading the material.

A competitive element will be introduced into the scenario to make

the students more motivated to use the application and interact with it more than just one time. The topics (learning goals) that are the focus for this scenario will now introduced. The topics are the foundation for the e-learning environment and the design and implementation will be closely related. The learning goals for set for this scenario are :

- Genes and gene expression
- Microarrays
- Survival analysis

As stated in the selected study case, the focus is on the importance on DNA analysis and how important it can be to look at the genetic information on individuals for health related issues. Having this established, I will now come up with a sample scenario for the study case that the students will follow.

6.3 Introducing the scenario

I propose this as the scenario to be used:

"You are the local doctor. You get a lot of patients with different types of cancer. These patients are not clear cases and you have the choice between 2-4 potential subtypes of cancer that might fail them. You don't want to overload the specialists with the patients that you could treat yourself. Therefore, with your simple, futuristic analysis tools, knowledge about DNA, diseases and genes you will do a more thorough analysis on the patients that are ill. Your mission is to try to determine what subtype of the cancer they have and try to find the best treatment for them. If you are still unable to figure out what subtype they have, you can state that this must be a new sub-type

and send them to a specialist. This is expensive and time consuming but failing to treat a patient should be avoided as far as possible.

Therefore you should determine as many correct subtypes as possible and find the right treatment. Not all subtypes are easy to determine, and in some cases it can be better to send them to a specialist. You will be ranked on the following criteria; correctness of determining subtypes and finding the most optimal cures. ”

This scenario can be played out as an interactive "game" for the students. They are presented a problem and they will play the “future doctor” in the process of determining what is wrong with different patients and later deciding the best treatments for them. Using this scenario we can establish clear objective for the students and describe some approaches in how to create an interactive application where a student can achieve this.

The primary objective for the student is to help as many patients as possible and to do this as correct as possible. In order to do this, the student will have to find out for each patient what subtype it is and find the best treatment method for the selected subtype. These are the basic elements of what the student will be doing.

Before describing the interaction process and going into a details there are a few questions that should be considered that concerns the scenario.

6.4 Things to consider

The next section will discuss the interaction process in detail, this section is meant to give an overview on the design issues and how to design the application. One question that arise in this scenario

are: how do we represent the genetic information so the students can figure out the difference between subtypes and actually learn something from this? The following steps here will help answer these questions.

6.4.1 Identifying a disease

How can we tell the difference between subtypes of a disease (specifically cancer)? The students will have to use their knowledge on microarrays, genes and gene-expression and use the correct view when determining a disease or subtype. Two different approaches are illustrated for this, having slightly different learning objectives. For simplification, the two approaches will be called the high-level approach and the low-level approach.

Low level approach

- Looking on the lower level; chromosomes and DNA.
- Compare pieces of a chromosome or a gene to a normal chromosome/gene
- Compare pieces of a chromosome or gene with known sequences in viruses.
- Check what treatment method works best for a selected disease or try to come up with one themselves

Learning objectives:

- Understand how important it can be to look on the DNA. Understanding genes and how mutations occur in a gene. See how deletions, deletions and point mutation can affect (or have no effect) on some of the mechanisms in the body.

- Understand polymorphic sites (snips) and how one amino acid can be more lethal than another under special conditions.
- See how viruses can insert DNA (and new diseases can derive from another) and how one could potentially cure it (by cutting out pieces of DNA or disabling a gene)

This approach has a lot of potential and could be developed to include protein structures and include other learning objectives. This approach is fairly complex and has elements from biology that is still not quite understood. It would require a lot more design if this approach were to be used. We will be using an easier approach, illustrating gene-expression on a higher-level basis. The high level approach which is described next, is far less complicated and is not focusing on the lower details on a gene.

High level approach

- Viewing gene-expression and using micro-arrays
- Seeing genes as a whole and expressed to a varying degree;
  - Micro-array representing important genes and their expression
  - And/or chromosome with genes marked and a color indicating the degree of expression
- Compare expressions and expression patterns to other diseases (or subtypes), using e.g. clustering methods
- Two ways of deciding a treatment;
  - Directly change expression of genes and observe the effect by evaluating statistical material presented (one option, but this will not be implemented)
  - Choose a treatment method based upon evaluating survival curves (This is the one we will use)

Learning objectives

- Understand how important it can be to look on DNA (comparing profiles, seeing similarities)
- Understand how genes can be expressed (different levels of expression)
- Learn to use statistical material presented in order to come up with a decision.

This example will not be suitable for diseases containing bacteria or viruses so such diseases will make little sense with this approach. Also, not all of the elements involved in this approach will be implemented (because of the lack of time), but the most essential ones such as genes and gene-expression using a microarray and seeing how microarrays can be compared (using clustering) will be implemented.

These two approaches share a lot of common data, but the learning goals and representation differ. Once one has been developed, adding the other approach (or study case using some of the material) will be much faster than creating the initial one because a lot of the code and material produced can be re used.

6.4.2 Selecting a disease to represent

This scenario will focus on determining specific subtypes of cancer. There are many types of cancer that could be applicable for the scenario (e.g. breast cancer could have been used). However, in order to introduce to students to something new that they might not be familiar with, Hodgkins lymphoma was chosen. This type of cancer attacks the lymphatic system of the body. The lymphatic

system is essential for humans, but it does not seem to be covered in the curriculum of high school biology. Therefore, by introducing Hodgkins and Non-Hodgkins lymphoma we could introduce the lymphatic system as a sub-goal that the students would learn about. Hodgkins lymphoma and Non-Hodgkins lymphoma are very different when it comes to diagnostics. Some of the sub-types of Non-Hodgkins lymphoma are really hard to distinguish. Microarrays can be used to create profiles and to compare the different subtypes. This makes Hodgkins (and Non-Hodgkins) lymphoma ideal for representing in the application.

6.5 The interaction - Step by step

The following section will describe the various steps involved in the scenario:

Introduction to the scenario :
The students are given a brief introduction to the scenario. The introduction should be small and concise to get the students interested in the application. After making sure that the students understand what to do, it will ask for their name (input) and the introduction screen disappears. The students are now operating as the future doctor "my name". The intention with this is so that one can later use this name when recording a score, and so the students could potentially compare with their classmates and have a small "competition" going.

Pre-learning process :
In order to achieve what the scenario asks of the students they are given some background information on how they can accomplish this. The application will provide a mean of reading about a given subject at any time. The students can choose to go to a main topic

and read about it in order to achieve their main goal. This application will introduce the students to the different topics used in the study case, and they can scroll through them and learn at their own pace. However, special consideration in how these topics are designed and presented must be taken. There is a need to represent them in an easy and fun way for the students, so avoid scaring them and to make it useful for the students so they actually learn something. Along with the topics, there should (where applicable) be interactive elements to aid the students in understanding the topic. The topics should also have illustrations and give hints to the students in how they can use this to accomplish the main objective of the scenario.

"The game" :

The interactive process here will run as a series of steps for each patient they get.

Selection process:

The students have a list of patients that are assigned to them. The patients will have real names to make it a bit more "realistic". Those patients that haven't been treated are selectable. When the student selects a patient, the application will show an automated match-comparison between the patient and the other subtypes. The representation of this is a visual comparison showing both the patients microarray and the disease microarray, along with a match-percentage. The student can click on a selected subtype to see how the clustering is performed and change the criteria for the clustering, or choose to diagnose the patient based upon the subtype it looks the most alike.

Suggesting treatment :

After selecting a diagnosis, the student will be asked to select a treatment for the patient. A Kaplan-Meier plot will be displayed

showing the survival curves for different treatments based upon the patients profile and the diagnosis suggested. The student is supposed to interpret the survival curves in order to find the best treatment. Also, if the student is interested in the underlying data for the Kaplan-Meier plots this can be accessed and displayed. When the student believes to have found the right treatment for the patient and has selected this, instant feedback will be provided to see if the diagnosis and treatment was correct.

Results and evaluation:
The "doctor" will be given the result of their treatment; whether it was successful or not, and an indication if they could've done better. As for the game the score is recorded and the current results are presented to the student. The student can then choose to treat more patients or quit the application.

6.6 Summary

Before implementation specific details are discussed, let's take one step back and summarize where we are at this point. The goal is to create an e-learning environment which purpose is to motivate students at high-school level for natural science subjects and give them an introduction to active fields in bioinformatics. In order to achieve this, I have come up with potential study cases that could be used and I have selected one for further development. The study case aims to show students how important it can be to look at DNA when it comes to health related issues. Furthermore, creating an applicable scenario has specified more details of the selected the study case. In this scenario the students using the application will be "playing a doctor", determining subtypes of non-hodgkins lymphoma and

finding treatment methods for patients. The goals for the students are reflected in what they will learn in the process. A discussion of different approaches on how to implement and achieve the goals has been made. The decision for this application is to go for one approach that is looking on the "higher level" of genetic information.

The scenario will introduce the students to the following topics: micro arrays, gene expression, Hodgkins lymphoma and survival analysis. In order to this, topics and the content surrounding must be designed so the students understand the underlying material. At this point the initial two phases (analyze and design) of ADDIE (see chapter 2) has been covered. The next chapters will cover the remaining phases and discuss the development, implementation and evaluation of the application.

Chapter 7

Implementation

This chapter will be discussing the choices that were made for implementation specific details of the application. It will discuss the alternatives there are for programming environments and languages. After selecting a programming language and environment, follows an overview of what tools, libraries and frameworks that was used in order to create the application. This chapter will show that there are many viable options to choose from when developing an e-learning environment such as this.

7.1 Criteria

Before choosing the programming environment some criteria for the application that will affect the choices later must be set. Below is a list of three important criteria that was considered for this application.

- **Interoperability**

  The application will be running on local school PC, with different hardware and operating systems. Therefore the application must be interoperable. It should run on most

common desktop operating systems (at least Linux, MacOS and Windows). The installation process should be minimal, no need for installing additional tools and libraries just to get the application up running. Getting the application up running should be easy.

- **Interaction**

  The importance of interaction in this application has already been stressed. The users should get responsive feedback in a valid amount of time and allow them to take part in the environment and actually see that something is happening. The interaction should be predictable (e.g. the result of an action should not come as a surprise to the users). The application should be fail safe (i.e. it should not crash because the user clicks on a certain button or does something outside the normal workflow).

- **Efficiency**

  Apart from immediate response from the interaction, the efficiency parameter is not very critical in this application. However the focus was on developing an application that didn't require to much system resources (not all schools have the newest computers with plenty of memory). Apart from this, the application had no criteria for the most efficient solution or implementation of an algorithm.

7.2 Tools and frameworks

This section will present a range of technologies, tools and frameworks that can be used for developing an e-learning environment such as this. During the remaining of this chapter, the application be-

ing developed will have sorted out what technologies and programming environment to use for the development process.

7.3 Choosing a programming environment

The first thing to be sorted out was what programming environment the application was to be developed in. In these days there are many good alternatives, so an introduction and comparison among the most popular ones was in place. The next section will go through these in order to select one of them.

- **.NET framework**

  This is one of the most popular development environments for developers today. The ease-of use and the integration with Microsoft Windows and native libraries ease the development process and makes a rapid development possible. The framework gives the programmer a choice between his favorite programming languages (Visual C++, C#, VB, Java.net). The .NET framework is based upon the original MFC idea (Microsoft Foundation classes) of wrapping parts of the windows API into C++ classes. .NET provides wrapper classes for all their supported languages so they all share the same features and API-calls.

  .NET uses a JIT (Just in time compilation) method or so called dynamic translation of the program. This allows for a richer yet slower language than pre-compiled languages (like e.g. C/C++).

  Microsoft provide an IDE (Integrated development environment) called visual studio for the .NET framework. This IDE is easy to use and has tools for rapid GUI-prototyping and debugging.

Unfortunately the .NET framework is designed for the Microsoft Windows platform and runs poorly on other platforms. There has been attempts to port MFC and the .NET runtime environment to other platforms (Linux, MacOS) [15] with varying degree of success. The .NET framework (with its core foundations) clearly runs best on Windows and is only guaranteed to work as intended on the Windows platform. Therefore, considering that this application is supposed to be a multi platform application, selecting the .NET framework would be a gamble. Crossing the fingers to hope it works with the Linux or MacOS port would probably not be the best of ideas. Choosing a better and more scalable environment would in this case be preferable.

– **C++**

C++ has been around since 1983 and has been one of the most popular programming languages for commercial programming [44]. It has a large community and a vast amount of libraries and gui toolkits. The source code is compiled to binary code specifically designed for the architecture, and then executed directly. This makes it efficient but it must be compiled differently for different architectures. However, it's still reasonable to make cross-platform C++ projects. In my point of view there are two issues why C++ is not optimal for this purpose. Most of the advanced gui-libraries in C++ that offer the different components needed for this application are platform specific. To keep my criteria about interoperability, one would then have to develop a separate "view" (GUI) for each environment the code was to run on[1].

[1]There are commercial, cross-platform C++ GUI libraries, but since they are commercialized they have been left out of this discussion

Apart from this, the C++ environment does not have a garbage collector (normally), and allows the programmer to do mistakes other environments prevent. Considering the time frame for this thesis, choosing C++ as a development environment would be time consuming because of the complexity of writing a robust, usable, multi platform code for an interactive application in this language.

- **Python**

  Python has become widely accepted in the programming community in the past years. It contains different GUI toolkits and a large amount of libraries for different purposes. One of the main advantages with Python is that it is an interpreted, dynamic typed language. Also, Python offers list-comprehension, object orienting and most of the features available in new programming languages. This makes Python excellent for rapid development and scripting tasks. The Python engine is delivered on MacOS, Windows and Linux platforms and the code is OS independent. However, since this is a purely interpreted language, it tends to run slow for larger application. Python is better suited for smaller applications and text-processing, not in large (potentially) interactive applications. Also, the GUI-frameworks provided in Python seem to be in the "early" ages. But surely, Python is an interesting alternative and as it is developed further and computational power increases, excluding it from this kind of development would be wrong. Also, in a situation where trying out a range of ideas, Python is ideal for rapid prototyping.

- Objective-C and the Cocoa framework

  Objective-C with the Cocoa Framework is an option unknown

for many developers. It was created by Apple Inc. in California and runs solely on MacOS.

Apple provides the following description of the Cocoa environment [21]:

Cocoa is an object-oriented application environment designed specifically for developing Mac OS X-only native applications. The Cocoa framework include a complete set of core classes and for developers starting new MacOSX only projects, Cocoa provides the fastest way to full-featured, extensible, and maintainable applications. Cocoa is one of the application environments for Mac OS X and has peer classes to communicate with Carbon and Java (Also application environment for Mac OS X). It consists of a suite of object-oriented software libraries and a runtime engine, and shares an integrated development environment with the other application environments.

The Cocoa framework was designed to be integrated with the objective-C programming language, but it is no requirement for a Cocoa application to be written in Objective-C. The objective-C language itself has most of the features that Java, C++ (and other OO-languages) provides. The language just went through a redesign [17] and got features that Java/C++ have had for a long time; GC (Garbage collection), dot-notation and other features that makes the language easier to use and more readable. The language itself is nothing exciting but the framework that is tightly bound to it is.

The framework aims at giving the the users a consistent look and feel for the applications and aids the developer in the process of complying with the human interface guidelines which Apple also provides. Apple is well known for their interfaces and the

usability and ease ability they offer in application design. The Cocoa framework integrates smoothly with the development environment on MacOSX (Xcode and Interface builder). The tools and the framework also follow the strong principle of a MVC design (Model, view, controller) with the higher level view-classes, core data framework for data persistence and the NSController class for controller parts. This design is the goal most developer should aim for because it enforces readability, reusability and maintainability.

In my point of view (after developing applications in both Eclipse (Java) and Visual studio (.NET) for more than 3 years), as an IDE (Integrated development environment), Xcode and interface builder is probably the best integrated and least painful to use for developers creating gui-applications.

As mentioned, the Cocoa framework also integrates with other languages (one doesn't have write Objective-C in order to use Cocoa). Apple has design so called "cocoa-bridges" to other languages (Java, Python, Ruby) so the application design and integration can be done separately and allows the programmer to choose his favorite language when writing the application logics. However, these bridges for other languages still lack some functionality compared with the flexibility of using Objective-C.

One major drawback is that the Cocoa framework is designed and will only run on the MacOS operating system. Considering that the number of Mac users is still low compared to Windows (and Linux) users choosing this as an environment for the application would not be ideal. However, the usability and the ease of development using this environment is worth noticing and when designing an interactive application, Apple knows

how to aid the programmer and to ease the development process.

- **Java**

  Java has been around for many years. It has a large community of developers, supports interoperability and flexibility of creating web-applications or standalone GUI applications. The idea with Java is that the source code is compiled into byte code and then runs on a virtual machine (Jvm, Java virtual machine) where the code is translated into instructions. The byte code solution Java uses makes it possible for the Java code to run on different machine architectures and makes the application OS independent [31].

  The Java community is large and getting help and support with Java questions is easy. Because of the open source community, there are many good, free and available IDEs provided for the Java environment.

Based upon previous experience with Java and considering the criteria set for the application, the choice fell on Java as the programming environment.

I'm not claiming that Java was the only reasonable environment for this application. If the application was not supposed to run on multiple operating systems (e.g. just run on Windows), Microsoft .NET would be ideal for obvious reasons. Unfortunately, .NET is only designed for Microsoft Windows and would run poorly on other systems [2]. Python as an alternative to Java in this case was strongly considered but because of lack of experience in Python development the choice fell on Java.

[2]There has been attempts to port the .NET framework to other platforms. Mono is such a port for MacOS and linux. It is still not not fully supported and is at its infancy

7.4 Gui-components

7.4.1 Choosing a gui-toolkit

A GUI toolkit is a set of building blocks for graphical user interfaces (GUI). They are often implemented as a library or a part of an application framework [49]. In general, they provide the programmer with a sophisticated API (application programming interface). This allows the programmer to design and create an interactive user interface reusing different components (widgets) provided in the API and customizing them as needed. This makes GUI development more bearable instead of having to deal with low-level system calls or native calls to render the graphics. There are many obvious advantages of using such a toolkit .

The GUI toolkit is the very foundation of every interactive application as for the one being developed. Hence selecting a good and well documented toolkit is important. There are many gui-toolkits available for Java, so this section will discuss the three common gui-toolkits used in Java and choose the one most suited for this application.

IBM provides an overview of the three gui-toolkits to list what features they support:

From this overview we see that both swing and SWT are equally good when it comes to features and the use of widgets. AWT on the other hand, lacks serious features and will therefore be left out from here on.

Choosing the best gui-toolkit from these two based on this overview alone would be insufficient. In order to reach a conclusion on what toolkit to use, a more detailed look into the two toolkits was needed.

Table 7.1: IBM: Comparison of GUI toolkits in java

Component	SWT	Swing	AWT
Button	X	X	X
Advanced Button	X	X	
Label	X	X	X
List	X	X	X
Progress Bar	X	X	
Sash	X	X	
Scale	X	X	
Slider	X	X	
Text Area	X	X	X
Advanced Text Area	X	X	
Tree	X	X	
Menu	X	X	
Tab Folder	X	X	
Toolbar	X	X	X
Spinner	X	X	
Spinner	X	X	
Table	X	X	X
Advanced Table	X	X	

The next section will introduce the two frameworks in brief, and I will look too see if there is any difference in how efficient and user friendly they are and see if any of them have any major advantage over the other. From here, a conclusion can be made on what toolkit to use for the application.

- **Swing:** Swing is the most common gui-toolkit used in Java. It relies on the AWT core, but is richer and responsible for its own rendering. Also, new in Java 5.0, it is customizable through XML (Javax.swing.plaf.synth). This makes swing a very flexible and suited toolkit for applications that requires customizability.
- **SWT:** "SWT is an open source widget toolkit for Java, which is designed to provide efficient, portable access to the user-interface facilities of the operating system on which it is implemented" [9]. SWT provides an interface for creating windows, handling user-interaction and do native GUI calls to the operating system it runs on. SWT replaces the normal gui-toolkit like awt/swing in Java. SWT uses native API calls to the operating system it runs on.

There are numerous articles on how SWT is more efficient and better than Swing and visa versa. I were unable to find consistent arguments throughout these articles and selecting between these two based on the arguments found seemed difficult. In addition to this, there are large support groups for both communities each claiming their toolkit to be better.

However, I managed to find a some of reasons for choosing Swing in favor of SWT:

- I had previous experience with developing Swing applications;

this would reduce the overhead of learning a new gui-toolkit. Instead I could refresh my knowledge on a familiar toolkit.

- Swing has been around longer and because of this have a larger community and more articles posted than SWT. Getting help or finding solutions to problems would probably be easier.
- Swing integrates smoothly with the IDEs and has a range of very functional WYSIWYG gui-builders. SWT has poorer support for such.
- Some claims that SWT look-and-feel will be inconsistent because of the native API calls to the operating system. If some advanced GUI functions are used, they will look and act differently on different operating systems. This argument seemed reasonable because of the different rendering methods in operating systems but would require further research to tell if this is a fact.

Apart from these arguments an implementation using SWT could be just as good but would in my case require more time learning properly and to do research on SWT. Because of the timeframe on this thesis SWT was dropped in favor of Swing.

7.4.2 JFreeChart

The study case to be implemented clearly needed a way to draw graphs or histograms (for survival analysis) . One option would be to develop own drawing methods for this, but that did not seem reasonable. There had to be libraries available that could provide such features. After looking around for a free open source library in Java I came across a promising library called JFreeChart.

JFreeChart provides this summary of their website of what their library provides [28]:

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

- a consistent and well-documented API, supporting a wide range of chart types;
- a flexible design that is easy to extend, and targets both server-side and client-side applications;
- support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);
- JFreeChart is open source or more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

This seemed suitable for this application. Since the application was not a commercial application, the LGPL licensing scheme was just fine and the features seemed to cover the needs.

7.5 Choosing an IDE

Most common IDEs are in these days Java compatible and they offer more or less the same features (by using plug-ins etc). In fact the process of choosing the IDE to use was based upon previous experience and personal opinions.

If however co-operating with more people in the development process this step should be considered more importantly, especially if the IDE doesn't support for project setups across different IDEs etc.

One could argue that one IDE is better than the other in terms of efficiency, support for various features etc. However the intention

with an IDE is to increase productivity for the programmer so in the end choosing an IDE is mostly personal opinions. I decided to use Netbeans, not because it is the best IDE but because it has the best WYSIWYG editor I have ever used for swing interfaces.

7.6 Summary

The application was developed using Java (1.5) and Swing (GUI library). For ease of development and a good WYSISWYG editor that integrates with swing, Netbeans was chosen as the IDE to use. An open source library called JFreeChart was used for creating and displaying graphs and statistical material. The next chapters will focus on the application design in detail and describe the user interface that was developed.

Chapter 8

Application design

As mentioned, one goal was to create a general framework for an e-learning environment that could be used in similar situations later and implement one study case in this framework. This chapter will come up with a design suited for the framework and the study case being developed.

First a brief overview of the system as the problem is viewed is given and an explanation on how the framework should interact with the specific scenario being developed. This chapter will cover the design of the framework and the implementation of the scenario. Note that most of the views will not be covered in this chapter as they are discussed in chapter 9.

8.1 General design

The first design is a conceptual design of the system. As seen in figure 8.1 there are two main parts of the application, the framework and the study case(s). The intention of this model is to show the basic structure of the application and how the two main parts are

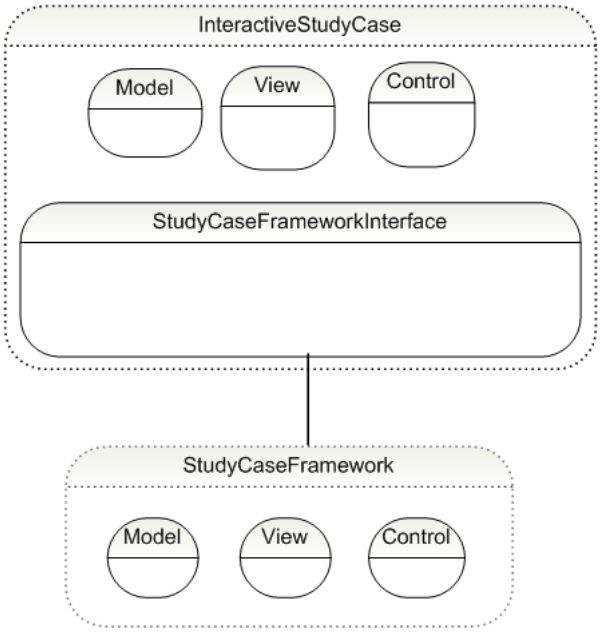


Figure 8.1: Conceptual design. See how the framework is connected to an interface. Both parts share their own view and controllers

connected. See that each of the two parts has its own MVC (model, view and control). This separation is not entirely true, in fact the actual implementation can share the model and some of the view developed for the framework.

This design allows for complete separation of the very basic operations needed for an interactive program to run. Also by using this conceptual design each study case can have its own MVC model independently. The study cases need to know nothing about the internal structure of the framework or environment it is loaded in. Each study case will conform to an interface with some common shared methods, but will be able to use its own model and view apart from this restriction. Generic components from the framework can be reuse. To see how the framework connected to a implementation is

intended to work see figure 8.2.

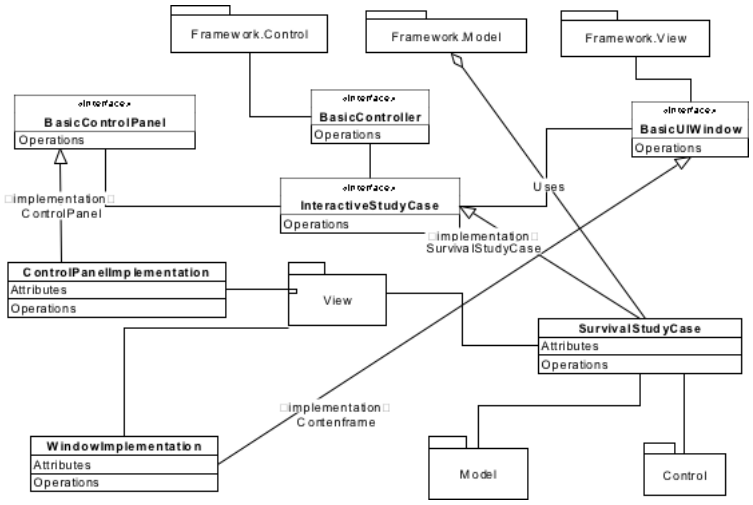


Figure 8.2: A more detailed model showing the relationship between the interface for a study case and the connection to the framework

Figure 8.2 illustrates how the framework is intended to work and how a study case is implemented using the framework. See how the implementation (SurvivalStudyCase) is able to re-use the data model from the framework. The views in the actual implementation are able to access the generic views from the framework. See how the window implementation and control panels implements the generic interfaces from the framework. Also a selection of components was developed in the framework and can be re-used in any implementation following this structure (see figure 8.3). This should give an idea of how the framework will work and how a study case will be implemented using the framework. The chapter will now go in more depth on the framework explaining some of the components involved. Then follows a discussion on how the actual implementation was designed. Further details on the specific views in both the framework and implementation (E.g. window

implementations and the components from the view) are discussed in chapter 9.

8.2 Framework

All study cases should be wrapped in the same interface (share a consistent look and feel). This means they share some common properties and are handled in a similar way when added to a user interface. To do this an interface that all study cases had to conform to were created. This interface is called InteractiveStudyCase as shown in figure 8.3. This figure also shows the other components involved in the framework. For simplicity the operations and attributes of the classes have been left out from the model, also the connection between models and view (E.g. How the MicroArrayView is connected to the MicroArray) is left out to reduce the complexity of the figure.

8.2.1 Model

The framework should be designed for study cases involving biology and bioinformatics. There are many model classes that could be included into such a framework. However, because there was limited time on developing this framework only the models needed in the implementation and for the potential study cases discussed were developed. Also, only the minimum of functionality required where included in these classes. From previous chapters we know that the application will contain microarrays, genes, clusters, hodgkins lymphoma, patients and survival analysis.

Microarrays, genes and clusters are quite generic for most study

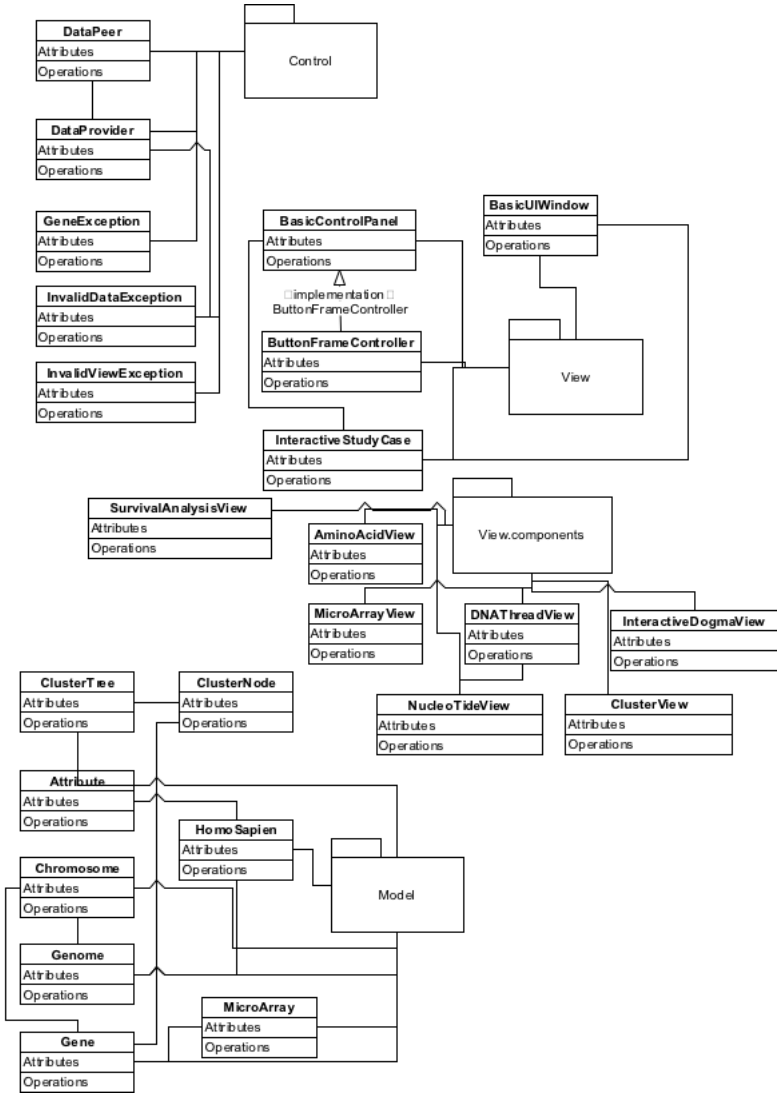


Figure 8.3: Figure showing the components of the framework and how they are connected to each other within the same package. Note that the connection between the packages (such as MicroArrayView and MicroArray) is left out from this model

cases. These elements resulted in the model of the framework. A patient is very specific for this scenario, but most of the surrounding functionality for a patient is generic. Hence, a more generic

class (HomoSapien) was introduced in the framework with possible extensions in mind. As we shall see later, a patient will extend this class. One could argue that to make this even more generic, e.g. using a classical taxonomy including inheritance from kingdom throughout specie and let Homo Sapiens inherit from this. This could be reasonable if a larger framework were to be developed. There is always the tradeoff between flexibility and extendibility and ease ability of development and maintainability.

The other classes involved in the model are not discussed as their names should be quite obvious and describe what they represent. For more information on these classes, I refer to the Javadoc created for the project[1].

8.2.2 View

The main intention for the framework view is to provide a consistent look-and-feel and avoid complexity when creating new study cases. Not having to create controller logics for handling windows and writing lots of code for handling GUI control (events, layout code etc.), makes the process in creating a new study case easier. The framework view components here aim to do some of this. The most important components in the framework view are:

- **InteractiveStudyCase** : An abstract class handling controllers and frames (loading and wrapping other classes).
- **BasicControlPanel** : An interface that all control panels will conform to in order to be added to an InteractiveStudyCase.
- **BasicUIWindow** : An interface that all content frames must conform to in order to be added to an InteractiveStudyCase.

[1] Javadochttp://heim.ifi.uio.no/~haraldf/master/Javadoc/

– **ButtonFrameController** : An abstract class providing a consistent look and feel for control panels. Functions for handling JButtons, and adding functionality for the framework (Loading frames when a button is clicked, disabling the current button, setting special actions and more.)

In addition to these components found in the framework view, there are interactive elements that was developed an put under this package. These are described in 9 where the user interface is introduced.

8.2.3 Control

The control logic in the framework is minimal. The control package contains a few exception classes (as shown in 8.3) for handling errors of various types. InvalidDataException, InvalidGeneException and InvalidViewException seemed natural to include in the framework (their names should explain their intention). Two generic classes for handling data storage and retrieval were added (DataPeer and DataProvider). These classes should be extended for a specific study case. The control package is by intention left minimal because it is up to each specific implementation to provide their own control logic.

8.3 The scenario

The scenario developed had some components that did not seem reasonable to place under the generic framework. These components will be discussed here. As shown in figure 8.4, the model classes Patient, Treatment and Disease were put under the study case design. The implementation has specific controllers (as most applications

have). One could argue that the model class for disease and treatment should be put under the generic framework. This was not done in the initial design and should be considered if re-factoring the code.

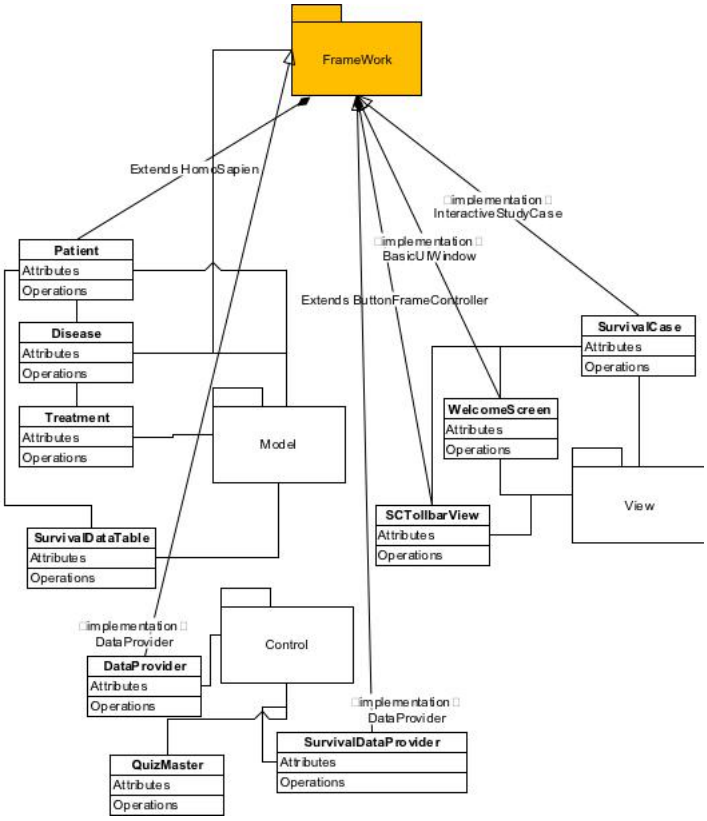


Figure 8.4: Simplified model showing the underlying design on the scenario in detail. Note that most of the view is left out from this model and is discussed in 9. See how the components of the implementation inherits from the framework.

Figure 8.4 leaves out most of the view as its components are discussed in chapter 9. See how the implementation uses and inherits from components in the framework. The following list describes the classes specifically used in the scenario:

- Disease : Representation of a disease. A disease is related to one or many genes (which is inherited from the framework). A

disease also has a list of treatments. This class contains functions for setting and getting possible treatments.

- Treatment : A simple data class representing a treatment.
- SurvivalDataTable : A data class that can be used in a JTable. It is responsible for calculating the estimated survival using the Kaplan-Meier estimator. It contains both censor and time data.
- Patient : Represents a patient in the system. A patient is linked to one or many disease (in this case one) and has functions to get and set diseases. Inherits functionality from the HomoSapiens class.
- QuizMaster : Controller for submitting answers in the user interface and handling this data.
- DataProvider : A controller that provides and manipulates data used in the application. Uses the SurvivalDataProvider for survival data. The controller developed only uses data that resides within the memory. No persistent storage method were used. This could be done by creating a more advanced DataProvider.
- SurvivalCase : This is the actual implementation of the InteractiveStudyCase interface defined in the framework. This class glues everything together and is essential for the application.
- SCToolbarView : Class responsible for the main navigation used in the application. Handles the topics to be available and how the navigation works. Inherits the functionality from the frameworks ButtonFrameController.

This should give an idea of how the application was implemented and how it is glued together with the framework. For simplicity, methods (operations) and attributes in the respective classes

have been left out. For those interested the source code can be downloaded from `http://heim.ifi.uio.no/~haraldf/master/source.tar.gz` and the Javadoc accessed from `http://heim.ifi.uio.no/~haraldf/master/Javadoc/`. The next chapter will cover the views and content frames in the application and framework as most of them has been left out from the discussion so far.

Chapter 9

User interface design

This chapter describe the user interface design and the content of the application reflecting the learning goals and objective set for the scenario. The elements described are found under the view package in the implementation. After describing these elements, section 9.8 will introduce the interactive elements developed for the framework.

9.1 General

The user interface strives to be as consistent as possible. This means that the buttons are located on the same spots on each screen, the text and the information is placed around the same locations. The user should get what they expect and no surprises (on the UI design). In order to make the objectives more clear and to separate the navigation from the actual content, the screen has been divided into three distinct parts:

- Topic navigation - Listing the main topics
- Sub navigation - Further dividing the main topics into distinct parts to reduce complexity of each screen

– Content frame - The actual learning material.

9.2 Constraints on user interface

The users are given a welcome screen at startup to give them some basic information about the program and how to use it. The user can continue using the application by clicking the first available topic. The user must go through each topic in a sequential order. This guarantees that the user has accessed a screen before going to the next. It also simplifies the requirements for the design of the content on each screen knowing that the user must have accessed a topic to get to the current screen.

9.3 Content

The content included in the user interface is closely related the selected scenario and the study case chosen. The content is viewed as the learning material and is tightly bound to what have been discussed in previous chapters. The next sections will briefly describe the material included in the application.

9.4 Navigation

The navigation consists of two parts; one for main topics and one for sub-topics. The top navigation contains a list of the main topics that the user will go through. Each of these topics is further divided into sub-levels. The sub-levels are displayed in a navigation frame on the left side. When the user changes the main topic the buttons with the sub-topics will change to match the current topic. There

is also a highlighting on the top navigation (current topic) and sub-level navigation (current sub topic) indicating where the user is and where to go. This is shown in figure 9.1 where the user has clicked on microarrays and is reading about computing.

The main topics included in the top navigation (as shown in figure 9.1) are [1]:

- **Genes and gene expression** - Containing the sub levels: "Genes", "Gene expression" and "Diseases"
- **Microarrays** - Containing the sub levels: "Microarrays", "Making microarrays" and "Analyzing microarrays"
- **Survival analysis** - Containing the sub levels: "Survival analysis", "Kaplan-Meier" and "Create your own plot"
- **Lymphoma** - Containing the sub levels: "Lymphoma", "Hodgkins lymphoma" and "Non-Hodgkins lymphoma"
- **Try yourself as a doctor** - As discussed in 6; diagnose a range of patients for subtypes of Non-Hodgkins.
- **Results** - Displays the results of the questions answered and diagnostics made.

When the user select a the main topics the sub navigation will display the sub-topics related. The learning material will be presented when the user selects a sub-topic. The learning material is included in the content frames which will be discussed next.

[1](Note that the actual user interface is in Norwegian, so the topics described here are approximate translations

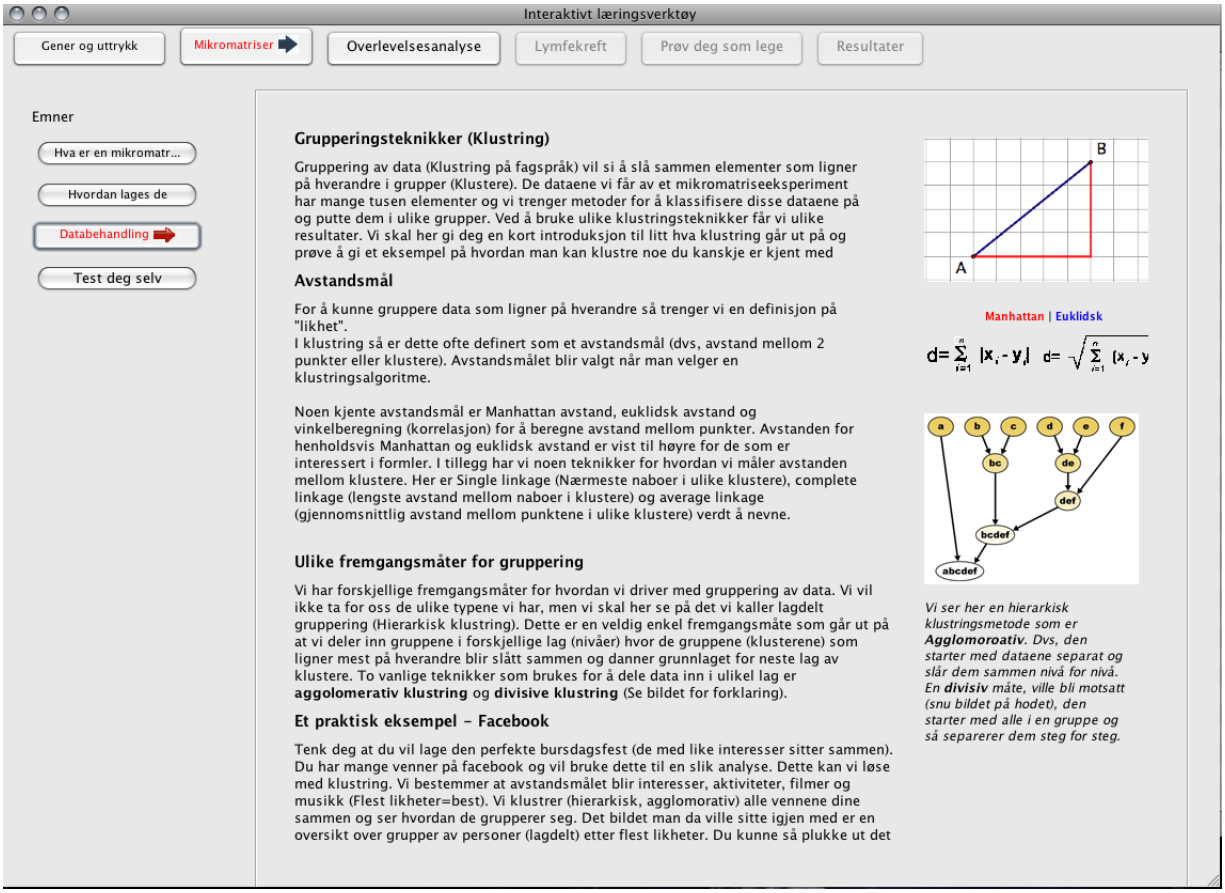


Figure 9.1: Figure shows the application in use. See the highlighting in the navigation indicating where we are and where to go next. The buttons on the top indicate the main topics in this application

9.5 Content frames

The third component making up the display is the content frame. The content frames contain the actual learning material and has components such as interactive elements, text and images. Figure 9.1 illustrated one of the content frames in the application, showing the learning material for microarray analysis. The content frames are always mapped to the same location when the user clicks on a topic

(or sub-topic). This enforces consistency and makes sure that the user knows what is going to happen when clicking a button.

The topics reflect the learning goals and objectives set out for the scenario. Below is a list given in a sequential order (flow of the application) for the different content frames used:

- **Genes** : Introducing genes, DNA and how the central dogma works and how genes are translated into proteins.
- **Gene expression** : Introducing gene expression and describing how genes can be expressed at various levels.
- **Diseases** : Briefly explaining that diseases can be related to our genetic material and emphasizing that it can be important to look at the genetic material when determining illnesses.
- **Microarrays** : Introducing microarrays and the technology
- **Making microarrays** : The process involved in making microarrays
- **Analyzing microarrays** : Showing how cluster analysis can be used in analyzing microarray data and giving a small example of clustering the students can relate to
- **Survival analysis** : Giving a brief introduction to survival analysis and how this can be used.
- **Kaplan-Meier** : Introducing the Kaplan-Meier estimator in an easy way leaving out statistical details.
- **Lymphoma** : Introducing the lymph system and cancer in this region. Some of the material included for this topic was taken from [7].
- **Hodgkins lymphoma** : Hodgkins lymphoma as a specific type of lymph cancer

– **Non-Hodgkins lymphoma** : Introducing different types of non-Hodgkins lymphoma

9.6 Consistency

The user interface developed should conform to Schneidersmans principles discussed in chapter 2. The constraints set on the user interface enforces dialogs that yield closure and groups similar functions together (such as the navigation for the topics). However, each content frame can be designed differently, so I have put an effort in trying to design these as consistent as possible. The reason for allowing the content screens to be designed differently is to have greater flexibility on how the material is presented and to making them easier to create (integrating them with the WYSISWYG editor in netbeans). If I wanted to create more consistent content frames (with more constraints) I could have created a more generic class that all contents screens used. This class could have functions for placing buttons, text-elements and images on the same locations, relative to each other and conforming to a set of rules. Such a generic class could be ideal for a larger project with more time for developing (because that the rules of a "consistent" ordering can be quite complex). If more content frames were to be included this could save time , because layout code-creation is repetitive and time consuming.

9.7 Application in use

This section will go through a few screens of the implementation to see the application in use.

Figure 9.2 shows the welcome screen introduced to the students. The

welcome screen gives a short guide on how to use the application. See how all the topics except "genes and gene expression" are initially locked, forcing the user to click the first topic. The system will unlock the next available topic in a sequential order as the user clicks the main topics. In this case when the user click "genes and gene expression" the topic "microarrays" will be unlocked.

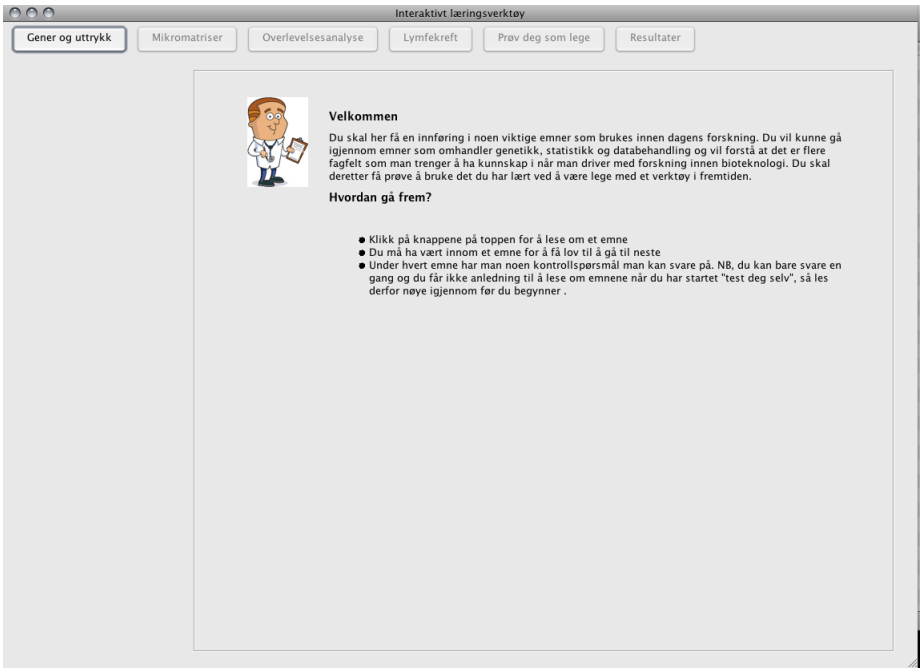


Figure 9.2: Welcomescreen. The first screen the students are introduced to

Figure 9.3 shows the sub-topics related to "genes and gene expression". Here the user has been through all sub-topics and decided to go back and read more about genes.

Figure 9.4 shows the scenario of this application. The student has been given 5 patients to treat. Once a patient has been treated the patient will be locked for further treatment.

In figure 9.5 the user has selected to diagnose a patient and has

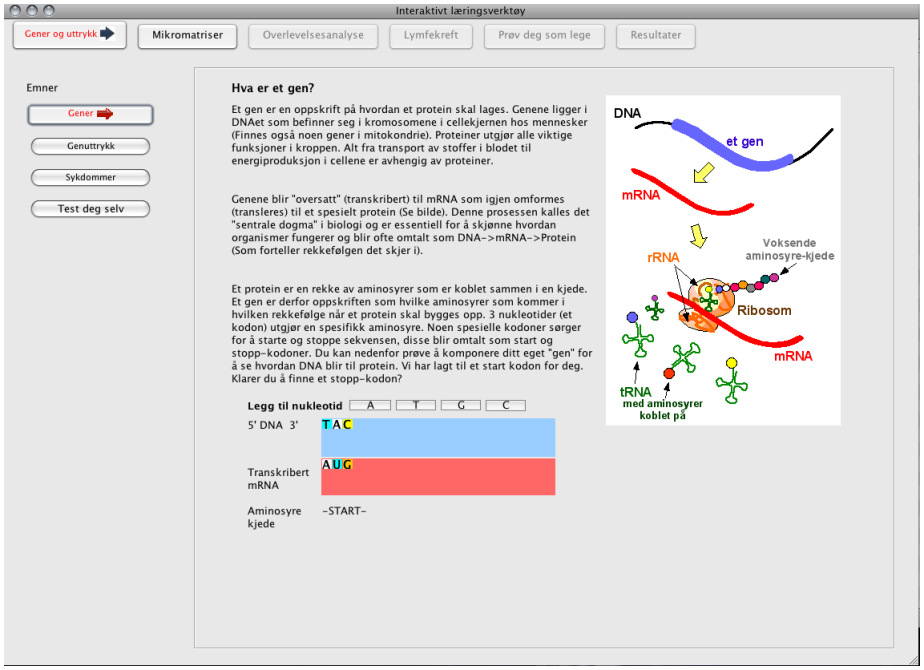


Figure 9.3: Displaying the sub-topic about genes. Note that one this screen the user has clicked through the sub-topics and decided to go back to the first topic.

decided to look at the clustering, comparing a patient and a sub-type.

This should explain how the interaction works and give an idea on what the application looks like. The next section will describe some of the interactive elements that is found in the content frames.

9.8 Interactive elements

The content frames in this application use interactive elements (visual components) that were developed for the framework view (see figure 8.3). This section will describe these components. Note that these components are also commented in chapter 10 and chapter 13 for reusability and future work.

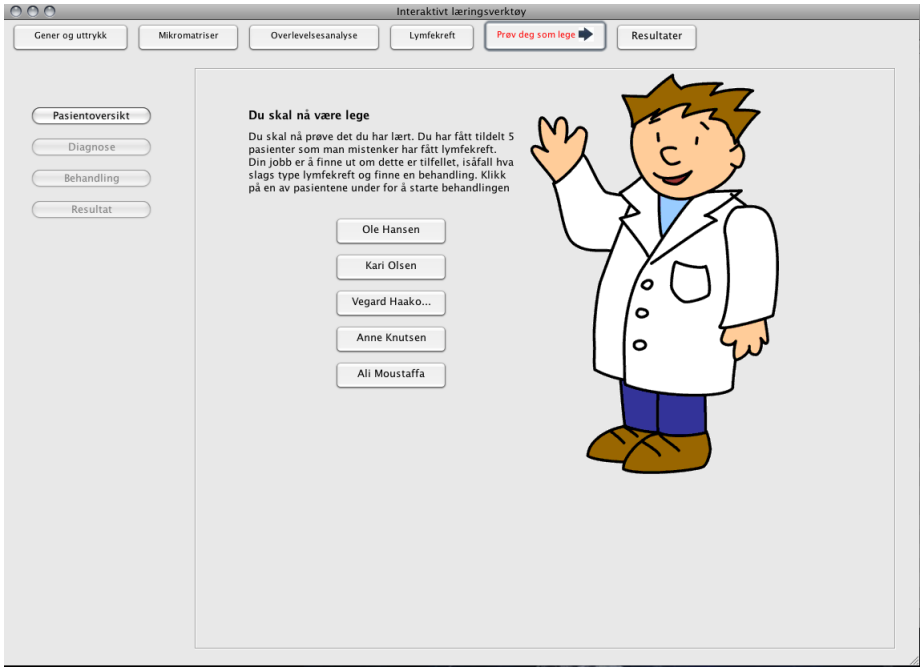


Figure 9.4: The scenario of the application. The user has been given 5 patients to treat

9.8.1 Microarrays

Microarrays are very essential for this application, so a component displaying a microarray with a number of genes in a similar manner to a normal microarray was created. Also an extension to this component was made that allow the user to click on a spot (gene) and adjust the sample and reference expression to see how the colors are affected by this change. This view could help some of the users to understand what a microarray image actually represents. Figure 9.6 shows the implementation of this component.

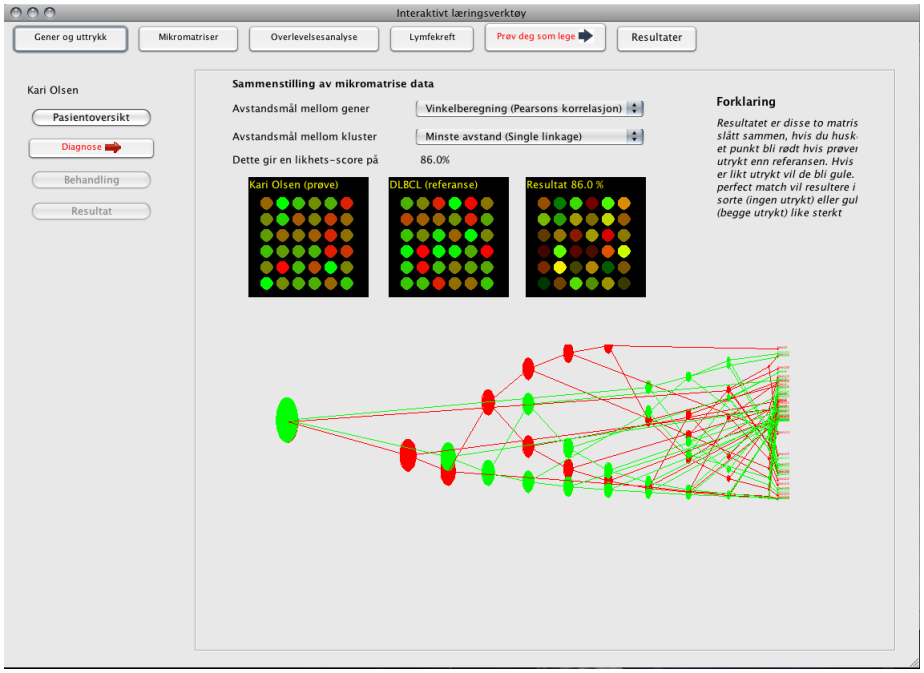


Figure 9.5: Advanced view showing the cluster for comparing a subtype of lymphoma to a patient microarray

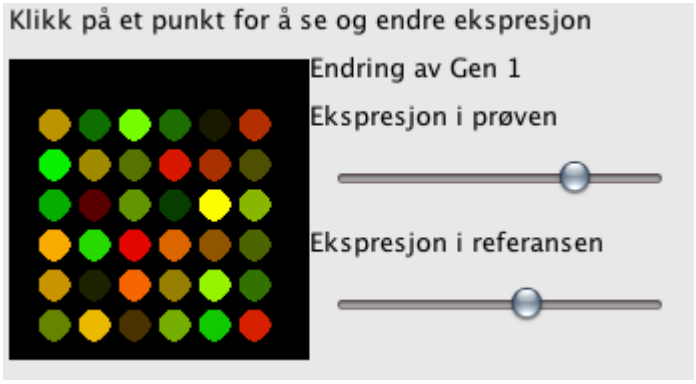


Figure 9.6: The interactive microarray that was developed. This figure shows the extended component where a gene (spot) has been clicked and allowing the user to change the expression

9.8.2 Interactive dogma

To illustrate how DNA is transcribed and translated into a protein, a small interactive component was created. This component allows the users to add nucleotides to a DNA and directly see the complementary mRNA transcribed. Also for each codon added (3 nucleotides) an amino acid is added to make an amino acid chain. If the user accidently finds the codon for a stop-codon, the next amino acids added are marked and a status message will be given in order to make the user understand the process of translation into proteins. The implementation of this component is shown in figure 9.7.

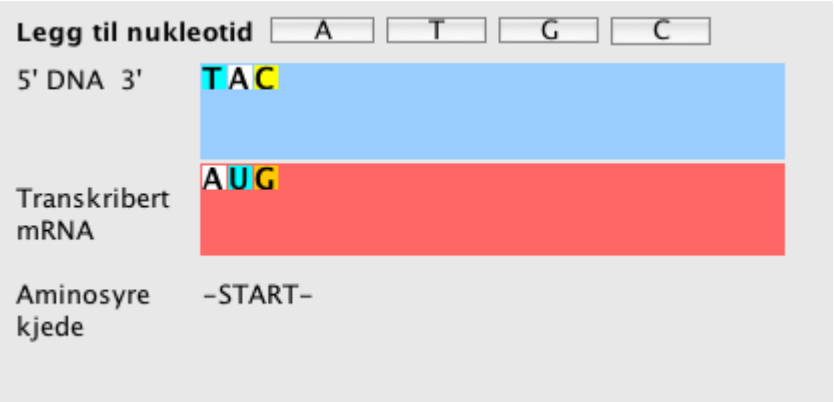


Figure 9.7: The figure shows the interactive dogma view that was developed. Here 3 nucleotides have been added; see how the amino acid chain is built matching the mRNA sequence.

9.8.3 Creating plots

A view that allows the user to create a graph by adding Kaplan-Meier plots and adjusting the times of event was created. The user can change the name, the censor states and the times involved in a plot. The user can add up to 5 plots in the same graph to see how it is

generated and manipulated interactively.

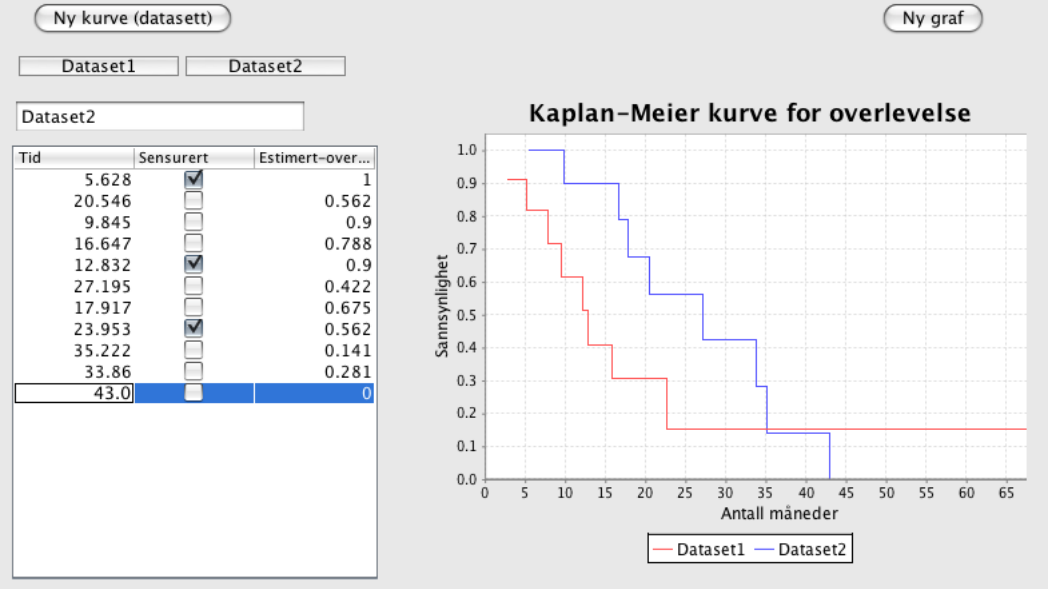


Figure 9.8: The figure shows the interactive plot view which allows the user to add data sets, change the labels and edit the data involved in each data set. Note that the graph is automatically updated to match the data sets involved.

9.8.4 Survival curves

For the survival curves related to a patient treatments, an option allowing the user to click to see what data is behind the curves was included. The user can also adjust the data (temporary) to see how it would affect the survival curves.

9.8.5 Clustering techniques

For the users that are interested in seeing how microarrays are compared or how the data is clustered, an interactive cluster

component was included. When the user trying to diagnose a patient can choose to click on a disease in order to make a cluster of the patients microarray and compare it to the selected disease. The user can select between the different distance method presented earlier and can select the linkage method between clusters. As the user changes these parameters, changes in the clusters are presented interactively. This component is illustrated in figure 9.9. This component has many ways of improving and will discussed in chapter 13.

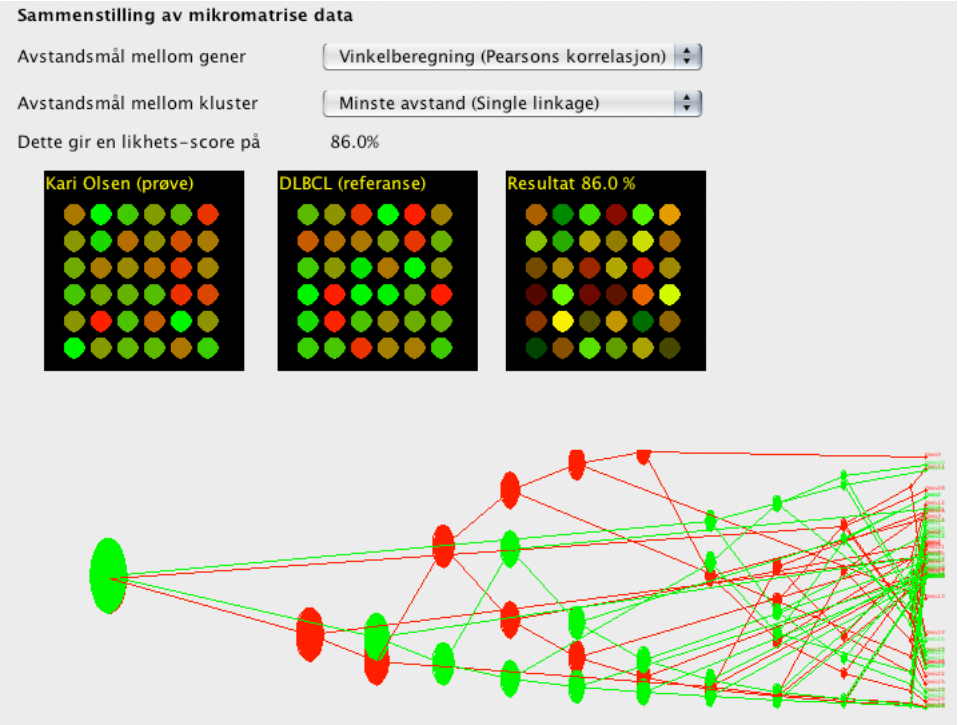


Figure 9.9: The figure shows the result of clustering two micro arrays and how they differ using a set of selected parameters for the clustering method. This is the interactive cluster view that was developed

9.8.6 Quizzes

Each of the main topics has a quiz. When the user clicks on the quiz-button, the sub-topics lock down (i.e. the user cannot access them). Each quiz has 3 questions that the users must answer in order to unlock the sub topics again. This is illustrated in figure 9.10. The lock-down is to prevent the users from going back and forth to read the correct answers and to make sure the users make an effort in remembering and reading the sections properly. When the users submit the quiz, a feedback is given. If the user answers wrong on some of the questions a red highlight on the selected option will appear to inform the user that the answer was false.

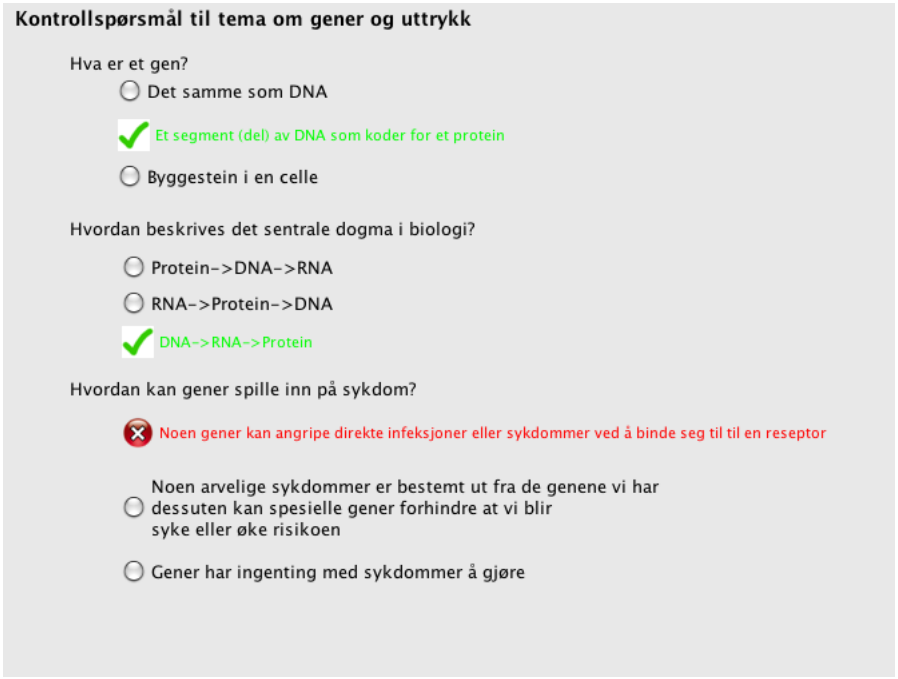


Figure 9.10: The figure showing the quiz-view after the user has submitted the answers. Correct answers are marked with green

Chapter 10

Reusability

This chapter will describe the important elements in the general framework that was made while developing the application. The idea is that some of these components can be used as a set for developing an e-learning environment taking a different approach or showing other topics. Implementation specific details and how the code logic is organized is not discussed here. The intention with this chapter is merely to show the purpose of the main components of the framework and give an idea on how to reuse them.

10.1 Application framework

10.1.1 InteractiveStudyCase - A GUI controller interface

This is the main GUI controller interface. It is responsible for handling the layout by wrapping content frames and control panels together. The intention with this interface is to keep a consistent look and feel throughout the application by providing the same load and unload functions for the content. The control panels

and windows that can be loaded will be extending the interfaces named BasicControlPanel and BasicUIWindow. Only a minimum of functions have to be implemented in order to load them into our controller and because of this, they are very customizable and flexible.

By implementing this controller we can re-use our content (and controls) but change the main look-and-feel on our application.

10.1.2 Controllers

DataProvider - Providing data to a study case

This is an interface that provides data for the InterActiveStudyCase. It has a bare minimum of functions and will for each study case be implemented slightly differently. Some study cases might want to connect to a database whilst others have their data stored in memory or on a flat file. The requirements for the data also differ therefore having a common data provider for all possible study cases would be impossible. Therefore this interface is as general as possible and must be implemented specifically for a study case.

10.2 Data containers - Models

10.2.1 SurvivalDataTable

This data class will take two sets of values, one set of time of events and one set containing sensor states and calculate the Kaplan-Meier estimator. This class extends the AbstractDataTable [10], which allows the data to be wrapper inside a JTable and the data to be edited.

10.2.2 Gene - An interface for representing genetic information

This is an abstract class for how a gene is represented. The MicroArrayView implemented assumes that a gene inherits from this class.

10.2.3 Microarray

This is a representation of a microarray in the sense of data. The current model has setters and getters for the genes involved in the microarray. Contain some operations to perform on the microarray data, such as calculating the average expression for rows or columns. This data class could also be extended to include much more information and functionality.

10.3 Components - Views

The components described here are interactive components (views) that were implemented and used within the content frames described earlier. These components can be re-used in different study cases. Note that the content frames can also be re-used but they are left out from this overview because they are very specific as they have been designed specifically for this application.

10.3.1 MicroArrayView

As described in section 9.8 a component for viewing a microarray was developed. This component is named MicroArrayView. The component takes a microarray data class and draws a microarray

based upon the expression contained in each spot of the microarray. The view can easily be extended to take in account actions when clicking or hovering a spot (e.g. display more detailed information of a gene when a spot is clicked). This is discussed in chapter 13. An extended version of this view was created [1] in order to illustrate how this is done and to make the process about reference and sample genes a bit clearer for the users. The user can click on a spot in the microarray and adjust the expression of the sample or the reference gene by using a slider and see how the color changes according to the expression.

When I was looking for pre-made packages to use to illustrate microarrays I came across a package called J-Express [36]. This package is fairly complex and is meant for scientific representation of microarray data. Using such a complex representation for microarray data and clustering would most likely be to difficult for the targeted group. I ended up with creating this component hoping it would prove useful and that it might be re-used in similar situations that needed a simpler representation of microarray data.

10.3.2 InteractiveDogmaView

This component aims to illustrate how the interactive dogma works. It allows for the users to add nucleotides to a DNA strand and see how the mRNA strand is built complementary to this. For each codon (3 nucleotides) added they would see a matching amino acid added to an amino acid chain. This component includes two other components that were made, the DNAThreadView and the AminoAcidView.

[1] This view is named InterActiveMicroArrayView and is located under the study case implementation

DNAThreadView

This component allows nucleotides to be added and removed and is built dynamically. If containing two strands, it will show the complementary strand in the same component. Also, this view can act as an mRNA display if the option is set. This component has potential in further study cases and could be included in an extended view such as the one discussed in chapter 13.

AminoAcidView

This view allows for adding of codons and builds up an amino acid chain with the right amino acids. If a stop-codon is added the next amino acids added will be grayed out along with a tooltip explaining that the translation will stop.

10.3.3 ButtonControlPanel

This is an abstract class for adding and removing buttons and for handling clicks on each button. Its intention is to group menus and behavior in a similar manner to reduce the complexity and to make it easier to change the layout of the entire application. The current implementation of this places a group of buttons in an horizontal or vertical layout and adds a step-by-step behavior and indication on the buttons as shown in figure 9.1. This is done in order to force a path through the application knowing that the users have been through certain screens before continuing to the next. The ButtonControlPanel allows for easy removal, disabling, adding and customizability of buttons for different screens without to much hazel.

Chapter 11

Evaluation of prototype

After getting the first prototype of the application up running I wanted to get someone to try the application in order to get important feedback and to find potential bugs that I had not thought of. The feedback was intended to give an indication of what needed to be fixed and improved before deploying the application on a larger crowd representing the target group. In order to to this, I created a little experiment. This chapter will describe this experiment.

11.1 Design of the experiment

The test group selected to try the prototype had to be chosen randomly. The test group should not have extensive knowledge of biology and computer science in order to have a more similar background to the target group. A pre-quiz and a post-quiz was to be given to the testers in order to gather data from the experiment and to evaluate the prototype. The intention with this experiment was to capture some of the problems and faults that were too obvious for me as the designer to see and to get feedback on how to improve

the application. The pre-quiz and post-quiz given to the testers are similar to those found in chapter 12.

11.2 Deployment

To make the test as random as possible and to make the odds of finding a test group matching the criteria I decided to go to the faculty of history at the University of Oslo. Although these people are not the same as the targeted group, chances were that they were similar in that had none or little knowledge of biology and computers.

To avoid selecting people by some bias I decided to take the first 4 people that walked through the main door after 12 O'clock (This was on a Thursday) and agreed to help by testing the application. A total of 6 people were asked in order to get 4 who had the time to help with this experiment. One could argue that 4 people is not sufficient for such an experiment. More people could probably provide more feedback and have different comments on how to improve the application. However, I limited the test to 4 people as I needed time to analyze the experiment and improve the application afterwards.

The testers were given a pre-quiz before trying the prototype. This was done in order to capture the knowledge they had in advance so this could be compared with answers of the post-quiz.

11.3 Feedback

A post-quiz was given to the testers after the test. A short dialogue with each tester was done in order to get additional feedback on the application.

11.4 Results of the experiment

Here is a summary of the data I gathered from this experiment.

Pre-quiz:

- None of the testers had biology background
- All (4 of 4) of the testers had heard of DNA and genes
- None of the testers had previous knowledge on the topic of microarrays.
- 1 of 4 of the testers knew that genes could be regulated without having further knowledge on the topic
- None of the testers had used an interactive learning tool in biology before.
- 1 of 4 had heard of lymphoma.

Post-quiz:

- 3 of 4 feels they learnt something new about genes and regulation.
- 3 of 4 says they have a clue of what a microarray now is.
- All (4 of 4) testers claim they now know what lymphoma is.
- 3 of 4 thinks that they have learnt a lot from the application
- 1 of 4 claims to have learnt "some" from the application
- 1 of 4 claims to be more interested in biology/bioinformatics after using this program, the rest claim that their interest did not change.
- 1 of 4 thinks the program worked well. The rest (3 of 4) thinks the program was OK.

- All (4 of 4) testers think it would be a good idea to have more interactive elements and to be able to click further into the subjects demonstrated.

Comments on the user interface:

- No feedback on question-forms. The users would like to know what question they answered wrong.
- A clearer indication of where we are at a given time in the user interface. The highlighting the buttons works for some extent but we need something in addition.
- Bad/Boring layout on some of the screen on Hodgkins-lymphoma.
- After one has taken a quiz on a main-topic and click the quiz-button again, the buttons lock (Bug / unwanted feature).

Comments on the topics:

- Clustering is somewhat difficult to understand.
- How genes/DNA work is not clear enough.

11.5 Evaluating the results

The post-quiz gave an indication that the testers did learn something about the topics introduced. So the application was not a total failure. The experiment also gave feedback for what elements needed improving to ensure greater learn ability and to make the application better. Having a dialogue with the testers about the user-interface and learning topics proved to be quite valuable in order to capture some of the problems that the quiz was unable to do.

11.6 Actions for improvements

The result of the experiment gave an idea for improvements in the application. A list of actions for ways of improving the application before a user test was created.

- Add an icon or better indication on the navigation items (Buttons)
- Add feedback when the user submits the answers on each quiz. This indicating that the selected answers are marked with a red or green color (red if the answer is wrong) and a small description describing why the answer is wrong.
- Clean up GUI / add some pictures and make it more interesting where it is possible.
- Go through the gene-topics and try to make it clearer
- Try to make clustering easier to understand. By adding better descriptions, and by making the interactive visualization better.

After improving the application with these actions, testing the application on the targeted group was at its place. The next chapter will be about the final user test. [1]

[1] The screenshots from the previous chapter was the result after improving the application.

Chapter 12

A user test of the system

12.1 Introduction

Having the main motivation for the thesis in mind I wanted to see if the application developed had any positive effect in getting the targeted group more interested in science (specifically bioinformatics) and in taking a higher level of education. In order to do this I created a little user test with the intention of capturing the user experience of the application and the topics presented. The user test was not meant to test whether the students gained any actual knowledge as this would be difficult to measure.

12.2 Design of the user test

The following list shows the questions I wanted to answer from performing a user test of the application on the targeted group:

- Did the students feel like they gained anything from using the application?

- Did the students have any previous experience with e-learning in biology?
- How well was each topic constructed? Is there any topic that felt less interesting for the students than others?
- Did the application make the students more interested in the selected topics?
- Is there any difference between sexes. E.g. are boys more interested in science than girls are? Did one of the sexes learn more from the application than the other?
- Is there a difference between students that have taken previous biology courses (2BI/3BI) and how much they learnt from the application?

To do this properly I needed a fairly large sample. In order to get a diversity of students, the application should be tested on different schools. Schools organize courses differently and use different learning material, therefore it was interesting to see if there was any difference between the schools and the user experience reported. These are the criteria set out for the user test:

- Large sample size (more than 50 students should be involved)
- Multiple classes and schools
- Test the application on students taking biology. There are second and third level biology classes on high-school, the test should capture if there is any difference between these levels.

In order to answer the questions, means of gathering data from the user test was needed. To solve this, a pre and a post-quiz (section 12.3) was created. The quiz was handed out in a paper edition to avoid confusing the students by including it in the application.

12.3 Pre/-Post quiz

Table 12.3 shows the pre-quiz given to the students prior to the user test. The questions intended to capture what the students knew in advance and map their motivation for science-subjects.

Table 12.3 shows the post-quiz that were given to the students. The post-quiz intended to capture what the students thought of the application in general, if they gained an interest for the topics and to measure the user experience.

12.4 Deployment of the user test

A few schools were contacted in order to find classes that were willing to help with the user test. As a result, the first two schools to give a positive response were used. These schools were "Oslo by Steinerskole" and "Risør VGS". A date for the user test was settled with the two schools. Also an installation of the application was coordinated with the IT administrators the day before the testing. This was done to avoid using time on installation and related problems that would affect the user test. In order to deploy the application efficiently and to avoid installing a java run-time environment on each of the computers, an executable java file of the application was created using Excelsior Jet [1]. The executable file can be downloaded from `http://heim.ifi.uio.no/~haraldf/master/interactive_learning.zip`.

On the day of the user test the pre-quiz was given to the students

[1] Excelsior Jet is an application that simplifies the creation of executable files of a java project. A shareware license of this application was used in order to create an executable file of the application developed

Table 12.1: Pre-quiz handed out to the students

Question	Type of answer
Sex	Single option, male/female
Background	Multiple options, 2BI, 3BI, 2MX, 3MX
Relates to the term bioinformatics	Single option, yes/no
Knows what a microarray is	Single option, yes/no
Heard of gene-expression	Single option, yes/no
Previous experience with e-learning tools in biology	Single option, yes/no
Think e-learning can be better than normal learning methods	Single option, yes/no
Grade of interest in studying science at college/university	Single option, ranging from 1 (not interested) to 5 (very interested)

Table 12.2: Post-quiz handed out to the students

Question	Type of answer
Overall feeling of the application	Single option, ranging from 1 (not satisfied) to 5 (very satisfied)
Increased interest for the topics	Single option, ranging from 1 (not increased) to 5 (Greatly increased)
Learning content of genes and gene-expression	Single option, ranging from 1 (low) to 5 (high)
Learning content of microarrays and computation	Single option, ranging from 1 (low) to 5 (high)
Learning content of survival analysis	Single option, ranging from 1 (low) to 5 (high)
Learning content in general	Single option, ranging from 1 (low) to 5 (high)

along with instructions on what to do. The post-quiz was handed out on a separate page, with the instructions that these questions were to be answered after testing the application.

12.5 Results

For those interested in the raw data from this user test, an excel file containing the reported data can be downloaded from `http://heim.ifi.uio.no/~haraldf/master/usertest.zip`. The data has been processed in order to answer the questions previously mentioned. This section will show the results of what the user test captured.

12.5.1 Pre-quiz

Table 12.3 shows the result of the pre-quiz given to the students. The numbers in this table are number of students that answered "yes" on the questions in the pre-quiz. A relative frequency for each group was calculated to see if there was any significant difference between the groups. The result of this is shown in figure 12.1. We see that less than 50% of the students had heard of bioinformatics and microarrays in advance. There was also a positive attitude towards e-learning environments and most of the students had previous experience with such. There seems to be a noteworthy difference between the groups and their knowledge of microarrays. 24% of the students at "Oslo by Steinerskole" compared to 11% at "Risør VGS" reported previous knowledge of microarrays. One reason for this can be different course objectives and learning material.

Table 12.3: Number of students that answered yes on the questions in the pre-quiz. A total of 66 students were involved in the user test

	Total	Relates to the term bioinformatics	Knows what a microarray is	Heard of gene-expression	Previous experience with e-learning tools	Think e-learning can be better
Females	37	17	4	13	24	34
Males	29	9	8	14	24	26
Risør VGS	29	14	3	9	20	28
Oslo by Steinerskole	37	12	9	11	28	32

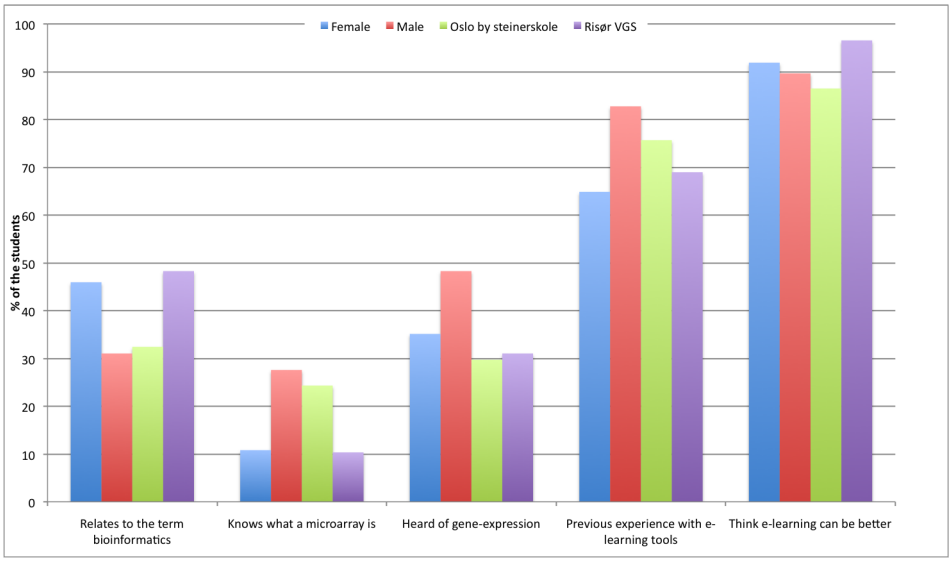


Figure 12.1: Histogram showing a relative frequency of data shown in table 12.3

12.5.2 Post-quiz

The data was grouped and average values were calculated. This was done in order to see if there was any difference between the groups in question and how useful the application seemed. The data is shown in table 12.4 [2].

Figure 12.2 show that the interest of continuing with science subjects after high school is lower for those that are taking 2BI than those taking 3BI. Also, the average values are lower on each question for the students taking 2BI compared to those taking 3BI. As we can see in table 12.4 the interest among male students are lower than for girls. To see how the options were distributed a list was created. This data

[2]Note that the question "grade of interest in studying science at college / university level" in this figure belonged to the pre-quiz, but for simplicity was included in the figure because of the quantitative options allowed

Table 12.4: Average values calculated for the qualitative questions showing the difference between different groups. Note that the values has been rounded up to the closest point decimal. A histogram of the exact data be found in figure 12.2

	Total	Risør VGS	Oslo by Steinerskole	2BI students	3BI students	Male	Female
Grade of interest in studying science at college / university level	3.2	3.1	3.4	2.6	3.9	3.1	3.4
Overall feeling of the application	3.8	4.0	3.7	3.7	4.0	3.8	3.8
Increased interest for the topics	3.7	3.7	3.7	3.4	4.0	3.7	3.6
Learning content of "genes and gene-expression"	3.8	3.9	3.8	3.5	4.2	3.9	3.8
Learning content of "microarrays and computation "	3.3	3.3	3.2	3.0	3.5	3.5	3.0
Learning content of "survival analysis"	3.6	3.5	3.7	3.5	3.7	3.7	3.5
Learning content in general	3.9	3.9	3.8	3.7	4.0	3.9	3.8

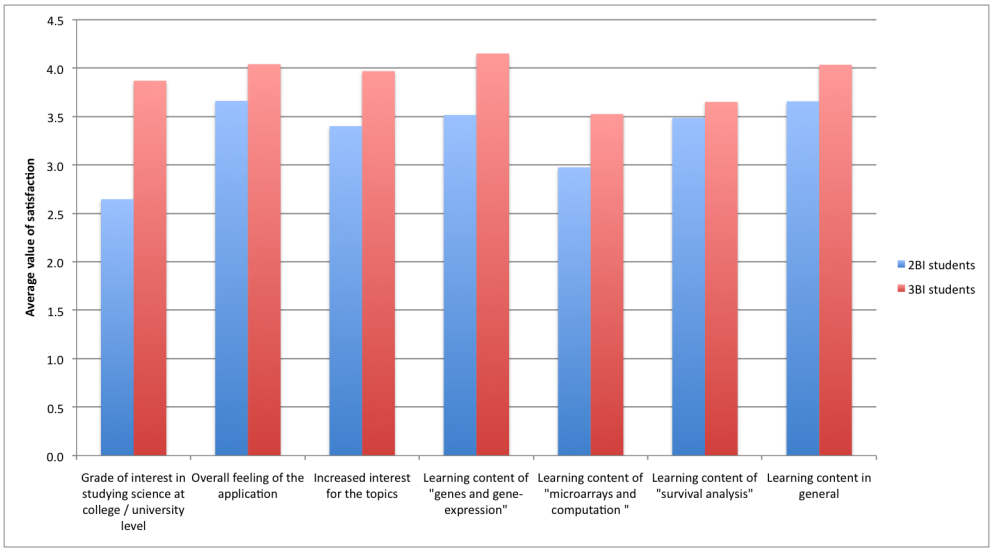


Figure 12.2: Comparing 2BI and 3BI students (see table 12.4). The scale ranges from 1 (low satisfaction) to 5 (high satisfaction). See that the average values are overall slightly higher for students taking 3BI.

can be found in table 12.5. A histogram showing this distribution is shown in figure 12.3.

12.6 Summary

The result of the pre-quiz shown in 12.1 shows that most of the students had previous experience with e-learning in biology (48 students of a total 66) and that they think e-learning can be better than normal learning methods. From figure 12.3 we see that 49 of the total 66 students selected the option 4 on the question about "learning content in general". This indicates that most students felt like they got something useful out of using the application. The figure also shows an indication that the application managed to increase the interest for these topics among most of the students (39 students

Table 12.5: Distribution of the options for the questions given to the students. Options ranking from (1=low) to (5=high)

	Option 1	Option 2	Option 3	Option 4	Option 5
Grade of interest in studying science at college / university level	8	9	19	17	13
Overall feeling of the application	1	2	14	41	8
Increased interest for the topics	1	3	18	39	5
Learning content of "genes and gene-expression"	0	4	19	30	13
Learning content of "microarrays and computation "	1	12	27	23	3
Learning content of "survival analysis"	0	5	19	40	2
Learning content in general	0	2	11	49	4

selected option 4). For the topic "microarrays and computation" the result seems to vary. The average values calculated in table 12.4 shows a lower average value for all groups on this topic compared to the other topics. This can indicate that the learning material or the design of the topic was difficult or hard to understand. There seems to be no significant difference between males and females and how much they learnt from each topic. When it comes to 2BI and 3BI students, the average values among the 2BI students for most of the topics are lower than with the students taking 3BI. One explanation to this is that the topics in the application might be a bit too difficult for students without previous experience of biology. As for the textual feedback some students gave, it seems like that there was too much reading material. One solution could be to make the theory more interactive or re-designing some of the frames, including less learning material per frame.

All in all, the application seemed to have a positive effect and hopefully inspired some of the students to continue with science subjects after finishing high school.

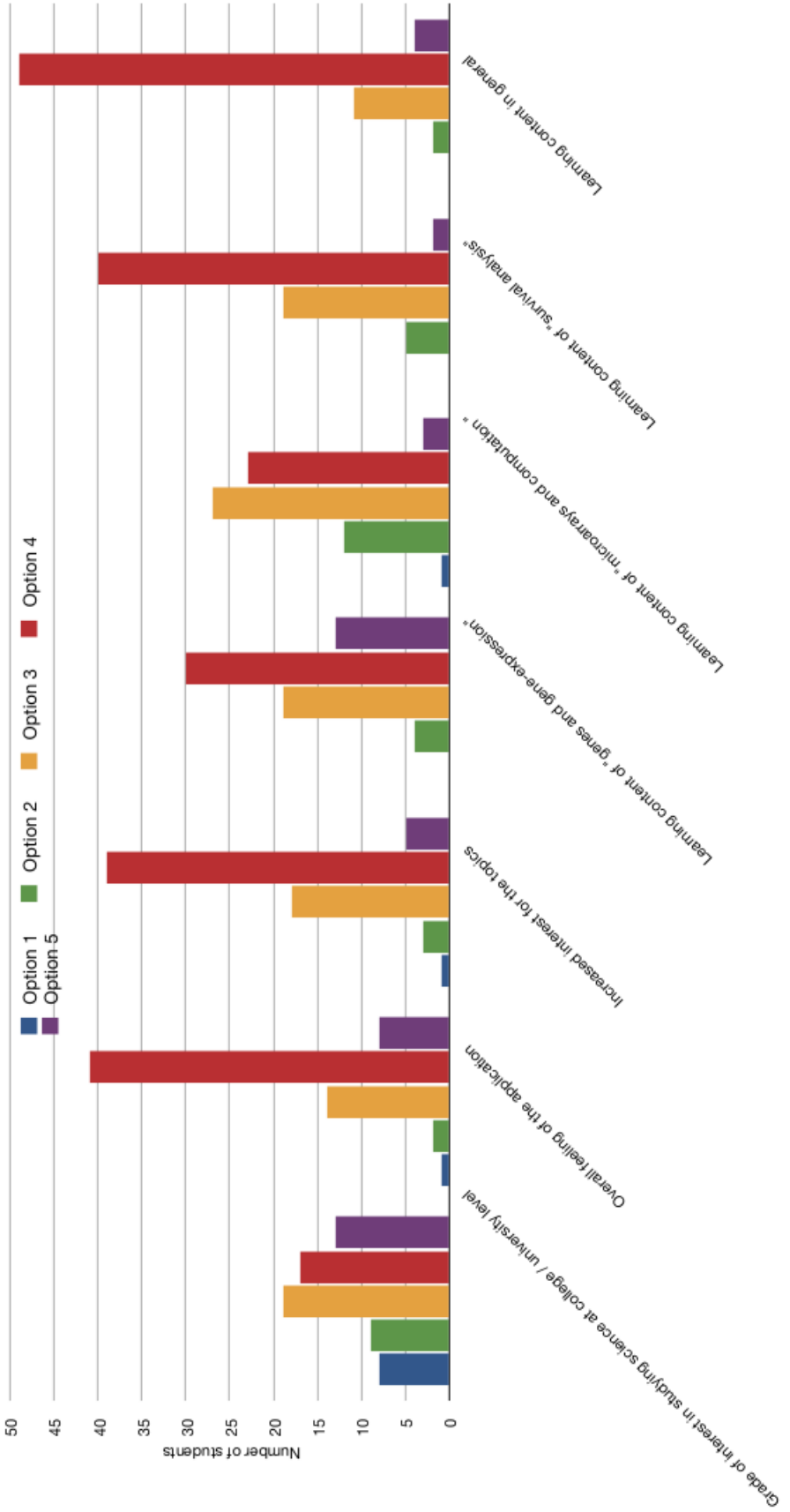


Figure 12.3: Histogram showing the data in 12.5. Each color indicates a distinct option

Chapter 13

Conclusion and future work

13.1 Conclusion

I have in this thesis presented an e-learning application aiming at high school students. Topics from biology, statistics and bioinformatics have been presented in a simplified manner to make the students interested in the selected topics. The user test performed indicates that more than 50% of the students feel they had learnt something from the application and that it increased their interest in the topics presented.

The user test also shows that most of the learning material could have been designed slightly better although the topics (except microarrays) seem to be satisfying (average satisfaction of more than 3.5 out of 5). The topic on microarrays on the other hand varies more on how much the students feel they learnt. It is hard to say whether this is a design flaw, bad learning material or that the topic is to difficult.

If starting the thesis over again with my current experience, the application would have been developed using a different approach than the ADDIE approach (see chapter 2). An iterative approach

(like an agile system development method) could have improved the learning material and the design of the application. This would require to have different high school students available for testing and feedback and to have contact with the targeted group early in the development process. I would also go through the learning material on the topics under microarrays (and analysis) too see how this could have been done different in order to make it better and easier for the students. Maybe focusing the entire application on just this topic (E.g. creating more interactive elements supporting the understanding of microarrays) would be better than the current solution.

For the implementation, a web application would be developed instead of the stand alone application developed. As the content in this application was quite simple, it would be considerable easier to develop this application as a web application and getting the application up and running on the different schools would be a lot easier. Developing and designing the general framework could have been done better. The resulting framework contains only a minimum of functionality required for the study case to work. A thesis aiming for designing and developing a flexible and general framework for similar applications could be ideal.

As for the user test, testing the application on other target groups would have been interesting. E.g. to see if the application had the same effect on students not taking any natural science related subjects or students at university level.

All in all, the application developed seemed to have had a positive effect. The students were overall satisfied with the application and the learning content and hopefully it recruited some of them for further studies in bioinformatics or natural science.

13.2 Future work

This chapter will cover some ideas for further development and improvements that came up during this thesis.

13.2.1 Improvements

Most of the material produced can be improved. This section will discuss some of the ideas of improvement that came up while developing the application and after deploying the user test.

Framework

There are many possible improvements of the framework. Examples include making the components more generic and adding more layers of abstraction. Adding more complex logic into the model classes (such as the genome and chromosome) could be useful for many different study cases. Examples of such extensions could be functions for searching and comparing genomes (or parts), manipulating genes or extracting other useful genetic information.

Content

The content screens can easily be improved. The user test in chapter 12 indicates an overall satisfaction with the learning material but that there is room for improvement. The screens about microarrays and clusters should be improved as students found this difficult to understand.

Interactive elements

Some of the interactive elements (components) that was developed could be improved in many ways. Here is a list of suggestions of what could be improved and some possible extension:

- **Clusterview** - This view has a lot of potential for improvements and extensions. One extension would be to develop multiple representations of the generated cluster. This could be a normal dendrogram, a visual map of some sort or perhaps a step-by-step cluster view. Letting the user go through each step and see what clusters merge next, could help in understanding clustering methods. Adding animations would make it more dynamic to the user. For the current representation in this application one extension would be to let the user click on the cluster (represented by an colored oval) and see what genes are involved in that cluster. By doing this, the user could click through the generated clusters and get an understanding of how clusters are generated.
- **Interactive dogma** - Some minor improvements in how this component acts and look should be considered. Adding pictures to see how the complementary nucleotides "connect" or glue together could help the understanding of complementary bases in the DNA threads. Also an explanation on the amino acids translated from the mRNA could easily be added and aid the understanding.
- **DNAThreadView** - The DNAThreadView that was developed is really simple. The current implementation only shows the nucleotides added. However, this view can be extended to include much more information. One extension would be to

include genetic information related. Maybe let the user scroll through the DNA and see what genes are at the current position, what are exons, introns and other information in the DNA viewed.

13.2.2 Ideas for new study cases

This section will briefly mention some of the ideas that seemed interesting but were left out because of the time frame. The target group is the same as before.

UV-radiation and its effects on DNA

The idea with this study case is to allow the users to see how UV-radiation affects the DNA (e.g. mutations) and to illustrate that there are reparation enzymes involved in fixing the DNA when damaged.

The student will be presented a person being exposed to the sun. The student can adjust parameters and see how this affects the person. The parameters chosen should be something the student can relate to. This could be applying sun lotion of different factor, adjusting the strength of the sun light or maybe decreasing the strength of the ozone layer. The user should get feedback on how damaging the sun can be for the DNA (Mutations, deletions etc), and see that there are enzymes continuously working to repair the DNA.

This study case could involve a lot of processes around DNA, reparation enzymes and mechanisms in the DNA and could be interesting because most people can relate to such an example.

Inheritance of genes through generations

In this study case the users will be able to learn about how genes are passed through generations, they could go in details and see how the genetic disease is linked and what differs from a normal individual.

One way this could be done is to let the student assign properties (such as genetic diseases, hair color etc) to individuals and perform a crossing to see how this is passed on to the next generation. A disease can be sex-linked, recessive (or dominant) and such a study case could aid the students in understanding inheritance. The study case could be designed to illustrate the basic rules of inheritance or to show more complex topics (such as imprinting, crossing over, polygenetic traits).

One of the teachers (Ingebreth Østegren) at the school where I deployed a user test informed me of a good interactive application that had some of these features, but the company developing it was shut down and did not distribute nor develop the application any more. I only got to briefly look at the software, but it could be interesting to look at this software and see if it is possible to re-use or extend it further.

13.2.3 New interactive elements

This section will go through two new interactive elements that could be developed and used in different situations.

Cell interaction

This interactive element would be suitable for the explaining cells and how they work. The idea is a feature where the users could

zoom into a cell and look at its components. There could be different levels of details being displayed (zoom-level). From a higher view the user could be looking at the overall process within a cell (e.g. transportation, looking on the cell-membrane etc). A more detailed view on the lower level could be looking at the transcription process and translation of mRNA into proteins. One possibility would be to let the user manipulate a gene and zoom out to see how this would affect the protein production and gene expression (and possible the organism the cell belonged to). The detailed view (looking at DNA) could be an extension of the DNAThreadView described in chapter 10. This feature has many possibilities and different learning goals and would need a proper design if being implemented.

MicroArray interaction

This interactive element would aim to explain microarrays further and allow the users to interact in the process of creating a microarray. Examples of such interaction is where the user selects the sample and reference cell used. They could then go through the process involved in creating a microarray (extracting cDNA, labeling etc.). This would hopefully aid in understanding how microarrays are created.

Further extensions would be to allow the user to interact with a gene and see how the expression would change on the microarray. Displaying useful information when the user click a spot could help the user in getting the bigger picture and see how certain genes can be the related to a disease.

Bibliography

[1] Image showing a dendrogram. http://www.resample.com/xlminer/help/HClst/HClst.8.gif.

[2] Image showing average linkage. https://www.ucl.ac.uk/oncology/MicroCore/HTML_resource/images/average_linkage.jpg.

[3] Image showing complete linkage. https://www.ucl.ac.uk/oncology/MicroCore/HTML_resource/images/complete_linkage.jpg.

[4] Image showing microarray data. [Online; accessed 20-April-2008], Url: http://compbio.utmem.edu/MSCI814/Module10.htm.

[5] Image showing single linkage. https://www.ucl.ac.uk/oncology/MicroCore/HTML_resource/images/single_linkage.jpg.

[6] Image showing the process in creating a microarray. Perou CM et al., 2000.

[7] Learning material on lymphoma and hodgkins disease. http://www.pasienthandboka.no.

[8] Second life. http://secondlife.com/.

[9] *SWT: The Standard Widget Toolkit, Volume 1*. Addison Wesley Professional, 2004.

[10] Java api for the abstracttablemodel, 2007. Url: http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/table/AbstractTableModel.html.

[11] BlashField Roger K. Aldenderfer, Mark S. Cluster analysis. 1984.

[12] Bioinformatics.org. Bioinformatics faq. [Online; accessed 08-January-2007], Url: http://bioinformatics.org/faq/#definitions.

[13] J. Black and R. McClintock. *An Interpretation Construction Approach to Constructivist Design*, pages 25–32. New Jersey, Educational Technology Publications, 1996.

[14] Altman D. Bland M. The logrank test. 2004.

[15] Bill Bodine. Porting windows mfc applications to linux. 2004.

[16] Bradley P. Glacken J. Bohannon, H. and R. McKnight. Principles for designing online instruction. 2001.

[17] David Chisnall. Behind the scenes of objective-c 2.0. 2006.

[18] A. A. Alizadeh et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[19] National Center for Biotechnology Information. Microarrays: Chipping away at the mysteries of science and medicine, 2007.

[20] D.W. Hosmer and S. Lemeshow. Applied survival analysis: regression modeling of time to event data. *Wiley, New York, USA*, 1999.

[21] Apple Inc. Cocoa fundamentals guide. 2007. http://developer.apple.com/documentation/Cocoa/Conceptual/CocoaFundamentals/WhatIsCocoa/chapter_2_section_1.html.

[22] National Human Genome Research Institute. Protocols in creating microarrays, 2007. [Online; accessed 13-December-

2007], Url: http://research.nhgri.nih.gov/microarray/protocols.html.

[23] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[24] Ian Hamilton Jay Cross. Dna of elearning - a short history of e-learning technologies with six trends taking us into the future. 2002. Url: http://www.internettime.com/Learning/articles/DNA.pdf.

[25] Paul Kaplan, E.L.; Meier. Nonparametric estimation from incomplete observations, 1958.

[26] Terry T. Kid. Handbook of research on instructional systems and technology. 1:1–13.

[27] Shi Leming. Dna microarray (genome chip) - monitoring the genome on a chip, 2002. [Online; accessed 12-October-2007], Url: http://www.gene-chips.com/.

[28] Object Refinery Limited. Jfreechart overview, 2007. http://www.jfree.org/jfreechart/.

[29] University College London. Cluster distances, 2005. Url: https://www.ucl.ac.uk/oncology/MicroCore/HTML_resource/Hier_Linkage.htm.

[30] Ian W. McKeague and Sundar Subramanian. Product-limit estimators and cox regression with missing cause-of-failure information. *Scandinavian Journal of Statistics*, (25):589–601, 1998.

[31] SUN MICROSYSTEMS. The java hotspot performance engine architecture. 1999.

[32] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.

[33] Thomas Jespersen Lars Aagaard Mogens Duch, Maria L. Carrasco and Finn Skou Pedersen. An rna secondary structure bias for non-homologous reverse transcriptase-mediated deletions in vivo. *Nucleic Acids Research*, 32, 2004.

[34] Andrew W. Moore. K-means and hierarchical clustering. 2001.

[35] P. Nagarajan. An overview of bioinformatics. 17(2):4–8, 2004.

[36] University of Bergen. J-express package. `http://www.bioinfo.no/tools/descriptions/jexpress`.

[37] MRC Laboratory of Molecular Biology. Central dogma of molecular biology. *Nature*, 227:561–563.

[38] Samordna opptak. Sluttstatistikk for samordna opptak 2006. pages 4–5, 2006.

[39] Samordna opptak. Søkertall 2007, 2007.

[40] J. Petraglia. The real world on a short leash: The (mis)application of constructivism to the design of educational technology, 1998.

[41] B Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, 2004.

[42] Gerhard Skagestein. *Systemutvikling - fra kjernen og ut, fra skallet og inn*. Høyskoleforlaget, 2002.

[43] Derek stockley. E-joruney on e-learning, 2005. [Online; accessed 15-October-2007] Url: `http://derekstockley.com.au/eindex2aa.html`.

[44] Bjarne Stroustrup. *The C++ Programming Language, Special Edition*. Addison-Wesley, 2000.

[45] Wikipedia. Estimator, 2007. [Online; accessed 27-November-2007], Url: `http://en.wikipedia.org/wiki/Estimator`.

[46] Wikipedia. Image showing microarray, 2007. url-http://en.wikipedia.org/wiki/Image:Microarray2.gif.

[47] Wikipedia. Parameters, 2007. [Online; accessed 27-November-2007], Url: http://en.wikipedia.org/wiki/Parameter.

[48] Wikipedia. Survival analysis, 2007. [Online; accessed 20-November-2007], Url: http://en.wikipedia.org/wiki/Survival_analysis.

[49] Wikipedia. Widget toolkit, 2007. [Online;accessed 17-December-2007], Url: http://en.wikipedia.org/wiki/Widget_toolkit.

[50] Wikipedia.org. Image showing euclidean and manhattan distance. http://en.wikipedia.org/wiki/Image:Manhattan_distance.svg.

[51] Jin Xiong. *Essential Bioinformatics*. Cambridge University press, 2004.